



TRABAJO DE FIN DE GRADO

DESARROLLO DE UNA APLICACIÓN DE SEGURIDAD  
VIAL EN ANDROID

Autor

Félix Sánchez Sánchez

Tutor

Fernando García Fernández

OCTUBRE DE 2017



## *Agradecimientos*

A Elena, por confiar en mí, apoyarme  
y hacerme día a día mejor persona.

A Portillo por darme la oportunidad  
de hacerlo posible.

## ÍNDICE GENERAL

Capítulo	Descripción	Página
1.	INTRODUCCIÓN.....	9
1.1	MOTIVACIÓN.....	10
1.2	DESCRIPCIÓN DEL PROYECTO Y OBJETIVOS.....	14
2.	ESTADO DEL ARTE.....	16
2.1	ADAS.....	16
2.1.1	ACC (Adaptive Cruise Control, control de velocidad adaptativo).....	16
2.1.2	Stop & Go.....	17
2.1.3	Asistente predictivo de eficiencia.....	18
2.1.4	Asistente de esquivas o anticolisión.....	19
2.1.5	Asistente de Giro.....	19
2.1.6	AEB (Automatic Emergency Braking, freno de emergencia automático).....	20
2.2	APLICACIONES MOVILES.....	21
2.2.1	Aplicaciones.....	21
2.2.1.1	Movon FCW.....	21
2.2.1.2	AcoDriver5.....	21
2.2.1.3	iOnRoad Augmented Driving 3.....	22
2.2.1.4	Mobile eye 2.....	23
2.2.2	Conclusiones.....	24
2.3	Técnicas para la detección y Clasificación.....	24
2.3.1	Técnicas de extracción de características.....	24
2.3.1.1	Descriptores tipo de Haar-like.....	25
2.3.1.2	SIFT (Scale-invariant feature transform).....	25
2.3.1.3	SURF (Speeded Up Robust Feature).....	25
2.3.1.4	HOG.....	26
2.3.2	Clasificadores.....	27
2.3.2.1	Aprendizaje no supervisado.....	27
2.3.2.2	Aprendizaje por refuerzo.....	27
2.3.2.3	Aprendizaje supervisado.....	27
2.3.2.3.1	Adaptive Boosting.....	28
2.3.2.3.2	Algoritmo KNN.....	28
2.3.2.3.3	Support Vector Machines (SVM).....	28
3.	DESCRIPCIÓN GENERAL DEL PROYECTO.....	29
3.1	CONTEXTO.....	29

## ÍNDICE GENERAL

Capítulo	Descripción	Página
3.1.1	LSI UC3M.....	30
3.2	PROPOSITO Y CARACTERÍSTICAS GENERALES DEL PROYECTO .....	30
3.3	DESCRIPCIÓN DE HERRAMIENTAS HARDWARE Y SOFTWARE.....	31
3.3.1	HARDWARE.....	31
3.3.2	SOFTWARE .....	33
3.3.3	Librerías OpenCv.....	33
3.3.4	Android Studio .....	34
3.3.5	SVMLight.....	35
4.	Descripción detallada de Funcionamiento .....	36
4.1	Aspectos Básicos de Android .....	36
4.2	Human Machine interface y LAYOUT .....	38
4.3	Aspectos Básicos de la Aplicación.....	39
4.4	Opciones de La Aplicación .....	39
4.5	Algoritmia para La Detección.....	40
4.5.1	Support Vector Machine.....	40
4.5.2	Configuración del detector basado en HOG .....	43
4.5.3	Aplicación para la detección de ciclistas.....	49
4.6	Metodo de Seguimiento y Persistencia Temporal .....	51
4.6.1	Filtro de Kalman.....	51
4.6.2	Algoritmo Hungaro.....	52
4.6.3	Sistema de estimación de Distancias. ....	55
4.6.4	Aplicación para el seguimiento de ciclistas.....	56
5.	Evaluación: Pruebas, Resultados y Limitaciones.....	60
5.1	Limitaciones.....	60
5.2	METODOLOGÍA PARA LA EVALUACIÓN .....	63
5.3	BANCO DE PRUEBAS PARA LA DETECCIÓN. ....	67
5.3.1	Resultados para el entrenamiento para ciclistas vistos desde atrás o de frente. ...	68
5.3.2	Resultados para el entrenamiento para ciclistas vistos de perfil.....	69
5.3.3	Resultados para el entrenamiento para ciclistas vistos de perfil y frente. ....	70
5.3.4	Conclusiones para la detección.....	71
5.4	Banco de Pruebas para el Seguimiento.....	75
5.4.1	Resultados para el Seguimiento.....	77
5.4.2	Conclusiones para el seguimiento.....	82
6.	PRESUPUESTO.....	83

## ÍNDICE GENERAL

Capítulo	Descripción	Página
6.1	MANO DE OBRA .....	83
6.2	MATERIAL.....	83
6.3	LICENCIAS DE SOFTWARE .....	84
6.4	PRESUPUESTO FINAL .....	84
7.	CONCLUSIONES .....	85
8.	TRABAJOS FUTUROS.....	86
9.	Referencias .....	87

## Tabla de Ilustraciones

ILUSTRACIÓN 1-1 ESTIMACIÓN DE CRECIMIENTO DEL PARQUE AUTOMOVILÍSTICO AUTÓNOMO [1]. ....	10
ILUSTRACIÓN 1-2 DATOS ACCIDENTES DE TRÁFICO ESPAÑA 2015 [3] .....	11
ILUSTRACIÓN 1-3 EJES DE ACTUACIÓN PLAN ESTRATÉGICO DGT [4] .....	12
ILUSTRACIÓN 1-4 DESGLOSE DE COSTES POR ACCIDENTE DE TRÁFICO [5]. ....	13
ILUSTRACIÓN 1-5 TAXI Y AUTOBÚS AUTÓNOMO.....	14
ILUSTRACIÓN 2-1 EJEMPLO DE FUNCIONAMIENTO DE ACC DESARROLLADO POR TOYOTA. ....	17
ILUSTRACIÓN 2-2 ACC. STOP & GO [8].....	18
ILUSTRACIÓN 2-3 ACC -ASISTENTE PREDICTIVO DE EFICIENCIA DE AUDI .....	19
ILUSTRACIÓN 2-4 ACC -EJEMPLO DE SITUACIÓN DEL ASISTENTE DE GIRO [13].....	20
ILUSTRACIÓN 2-5 ACC -EJEMPLO DE ACTUACIÓN [15].....	20
ILUSTRACIÓN 2-6 APLICACIÓN MOVON FCW [17] .....	21
ILUSTRACIÓN 2-7 APLICACIÓN ACODRIVER 5 [18] .....	22
ILUSTRACIÓN 2-8 APLICACIÓN ACODRIVER 5 [19]. ....	23
ILUSTRACIÓN 2-9 APLICACIÓN MOBILE EYE 2.....	23
ILUSTRACIÓN 2-10 HOG APLICADO A IMAGEN DE BIBICLETA.....	26
ILUSTRACIÓN 3-1 APP BASE LSI. ....	29
ILUSTRACIÓN 3-2 IVVI UC3M .....	30
ILUSTRACIÓN 3-3 HW XIAOMI REDMI 3 .....	31
ILUSTRACIÓN 3-4 HW XIAOMI MI5.....	32
ILUSTRACIÓN 3-5 ANDROID STUDIO .....	34
ILUSTRACIÓN 4-1 CICLO DE VIDA DE UNA ACTIVITY.....	37
ILUSTRACIÓN 4-2 ASPECTO DE LA APLICACIÓN DESARROLLADA.....	38
ILUSTRACIÓN 4-3 TRANSFORMACIÓN A UNA DIMENSIÓN SUPERIOR LINEALMENTE SEPARABLE. EL PROBLEMA DE LA BÚSQUEDA DE UNA FUNCIÓN DE DECISIÓN NO LINEAL EN EL ESPACIO DEL CONJUNTO DE EJEMPLOS ORIGINAL (ESPACIO DE ENTRADAS), SE PUEDE TRANSFORMAR EN UN NUEVO PROBLEMA CONSISTENTE EN LA BÚSQUEDA DE UNA FUNCIÓN DE DECISIÓN LINEAL (HIPERPLANO) EN UN NUEVO ESPACIO TRANSFORMADO (ESPACIO DE CARACTERÍSTICAS).....	41
ILUSTRACIÓN 4-4 HIPERPLANOS A DISTINTOS MÁRGENES .....	41
ILUSTRACIÓN 4-5 REPRESENTACIÓN DE HIPERPLANOS PARALELOS E HIPERPLANO MARGEN.....	42
ILUSTRACIÓN 4-6 EJEMPLO DE PREPROCESADO.....	44
ILUSTRACIÓN 4-7 EJEMPLO DE CÁLCULO DE BORDES. ....	44
ILUSTRACIÓN 4-8 DIAGRAMA DE ORIENTACIONES POR INTERVALOS Y REPRESENTACIÓN ACUMULATIVA POR INTERVALOS. ....	45
ILUSTRACIÓN 4-9 DIAGRAMA DE FLUJO DE LA APLICACIÓN PARA LA DETECCIÓN DE CICLISTAS. ....	49
ILUSTRACIÓN 4-10 EJEMPLO DE DETECCIÓN FRAME T. CADA CICLISTA REPRESENTA UNA DETECCIÓN Y LAS FLECHAS ILUSTRAN EL VECTOR VELOCIDAD Y SENTIDO. ....	52
ILUSTRACIÓN 4-11 EJEMPLO DE DETECCIÓN FRAME T+1. LOS CICLISTAS NEGROS REPRESENTAN LAS DETECCIONES NUEVAS PARA EL SIGUIENTE FRAME MIENTRAS QUE LOS CICLISTAS VERDES REPRESENTAN LAS PREDICCIONES PROPORCIONADAS POR EL FILTRO DE KALMAN.....	53
ILUSTRACIÓN 4-12 PUNTO DE CONTACTO ENTRE SUELO Y OBJETO. ....	55
ILUSTRACIÓN 4-13 ÁNGULOS DE EULER PITCH Y ROLL.....	56
ILUSTRACIÓN 4-14 DIAGRAMA DE FLUJO DE LA APLICACIÓN PARA EL SEGUIMIENTO DE CICLISTAS. ....	57
ILUSTRACIÓN -1 LIMITACIONES -PROBLEMA DE OPTIMIZACIÓN .....	60
ILUSTRACIÓN 5-2 LIMITACIONES -PROBLEMA DE OPTIMIZACIÓN .....	61
ILUSTRACIÓN 5-3 LIMITACIONES – CÓDIGO OPTIMIZADO CORRECTAMENTE.....	61
ILUSTRACIÓN 5-4 LIMITACIONES – DETECCIÓN CON FONDO E ILUMINACIÓN ADVERSA. ....	61
ILUSTRACIÓN 5-5 LIMITACIONES – DETECCIÓN CON FONDO E ILUMINACIÓN ÓPTIMA. ....	62
ILUSTRACIÓN 5-6 LIMITACIONES – PROBLEMA DE OVERLAPPING.....	62
ILUSTRACIÓN 5-7 MATRIZ DE CONFUSIÓN. ....	63
ILUSTRACIÓN 5-8 EJEMPLO ILUSTRATIVO DE LA MATRIZ DE CONFUSIÓN. ....	64

ILUSTRACIÓN 5-9 ESPACIOS CURVA ROC.....	65
ILUSTRACIÓN 5-10 CONSTRUCCIÓN DE LA CURVA ROC .....	66
ILUSTRACIÓN 5-11 IMÁGENES BANCO DE PRUEBAS (FRENTE) .....	67
ILUSTRACIÓN 5-12 BANCO DE PRUEBAS.....	68
ILUSTRACIÓN 5-13 IMÁGENES BANCO DE PRUEBAS (PERFIL).....	69
ILUSTRACIÓN 5-14 IMÁGENES BANCO PRUEBAS (PERFIL Y FRENTE) .....	70
ILUSTRACIÓN 5-15 BANCO DE PRUEBAS: EVALUACIÓN DE LOS DIFERENTES NIVELES DE UMBRAL PARA EL CLASIFICADOR DE DETECCIÓN DE CICLISTAS DE FRENTE EN FUNCIÓN DE LA SENSIBILIDAD Y LA ESPECIFICIDAD. PARA LA EVALUACIÓN DE CADA CLASIFICADOR SE EMPLEÓ EL BANCO DE IMÁGENES DE PRUEBA CORRESPONDIENTE Y 4 VALORES DE UMBRAL DISTINTOS. LA LÍNEA DELIMITADORA DIAGONAL REPRESENTA EL LÍMITE DE ALEATORIEDAD PARA LA EVALUACIÓN DEL CLASIFICADOR. EL PUNTO ROJO INDICA EL MEJOR RESULTADO OBTENIDO.....	72
ILUSTRACIÓN 5-16 BANCO DE PRUEBAS: EVALUACIÓN DE LOS DIFERENTES NIVELES DE UMBRAL PARA EL CLASIFICADOR DE DETECCIÓN DE CICLISTAS DE PERFIL EN FUNCIÓN DE LA SENSIBILIDAD Y LA ESPECIFICIDAD. PARA LA EVALUACIÓN DE CADA CLASIFICADOR SE EMPLEÓ EL BANCO DE IMÁGENES DE PRUEBA CORRESPONDIENTE Y 4 VALORES DE UMBRAL DISTINTOS LA LÍNEA DELIMITADORA DIAGONAL REPRESENTA EL LÍMITE DE ALEATORIEDAD PARA LA EVALUACIÓN DEL CLASIFICADOR. EL PUNTO ROJO INDICA EL MEJOR RESULTADO OBTENIDO.....	73
ILUSTRACIÓN 5-17 BANCO DE PRUEBAS: EVALUACIÓN DE LOS DIFERENTES NIVELES DE UMBRAL PARA EL CLASIFICADOR DE DETECCIÓN DE CICLISTAS DE PERFIL EN FUNCIÓN DE LA SENSIBILIDAD Y LA ESPECIFICIDAD. PARA LA EVALUACIÓN DE CADA CLASIFICADOR SE EMPLEÓ EL BANCO DE IMÁGENES DE PRUEBA CORRESPONDIENTE Y 4 VALORES DE UMBRAL DISTINTOS. LA LÍNEA DELIMITADORA DIAGONAL REPRESENTA EL LÍMITE DE ALEATORIEDAD PARA LA EVALUACIÓN DEL CLASIFICADOR. EL PUNTO ROJO INDICA EL MEJOR RESULTADO OBTENIDO.....	74
ILUSTRACIÓN 5-18 DEFINICIÓN DE TRAYECTORIAS EN AMBOS EJES.....	75
ILUSTRACIÓN 5-19 DEFINICIÓN REAL DE TRAYECTORIAS. ....	76
ILUSTRACIÓN 5-20 BANCO DE PRUEBAS: REGISTRO DE DATOS Y CAPTURA DE FRAMES. PARA LA EVALUACIÓN DEL SEGUIMIENTO Y EL SISTEMA DE ESTIMACIÓN DE DISTANCIAS SE UTILIZARON LOS FRAMES CAPTURADOS DURANTE LAS PRUEBAS Y UN REGISTRO DE DATOS. EL REGISTRO DE DATOS CONTIENE LOS SIGUIENTES CAMPOS:.....	77
ILUSTRACIÓN 5-21 ESTIMACIÓN DE DISTANCIAS A LO LARGO DEL EJE X Y SUS RESPECTIVOS ERRORES ..	79
ILUSTRACIÓN 5-22 ESTIMACIÓN DE DISTANCIAS A LO LARGO DEL EJE Y Y SUS RESPECTIVOS ERRORES ..	80

## Índice de Tablas

TABLA 5-1 INTERPRETACIÓN VALORES AUC.	66
TABLA 5-2 RESULTADOS VISTA FRONTAL Y TRASERA.	68
TABLA 5-3 RESULTADOS VISTA DE PERFIL.	69
TABLA 5-4 RESULTADOS MIXTO (FRENTE Y PERFIL).	70
TABLA 5-5 SENSIBILIDAD Y ESPECIFICIDAD PARA CLASIFICADOR DE FRENTE.	71
TABLA 5-6 SENSIBILIDAD Y ESPECIFICIDAD PARA CLASIFICADOR DE PERFIL.	71
TABLA 5-7 SENSIBILIDAD Y ESPECIFICIDAD PARA CLASIFICADOR MIXTO (PERFIL Y FRENTE).	71
TABLA 5-8 VALORES AOC OBTENIDOS.	74
TABLA 5-9 DISTANCIA MEDIA. DESVIACIÓN TÍPICA Y VARIANZA.	78
TABLA 5-10 ERRORES Y RATIOS DE DETECCIÓN.	81
TABLA 6-1 COSTES POR MANO DE OBRA.	83
TABLA 6-2 COSTES POR MATERIALES.	83
TABLA 6-3 COSTES POR LICENCIAS SOFTWARE.	84
TABLA 6-4 COSTES TOTALES.	84



## Abreviaturas y Acrónimos

ACC	Control Adaptativo de Crucero (Adaptive Cruise Control)
ADAS	Sistemas Avanzados de Ayuda a la Conducción (Advanced driver-assistance systems)
AEB	Frenada Automática de Emergencia (Automatic Emergency Braking)
AUC	Área Bajo la Curva (Area Under Curve)
DGT	Dirección General de Tráfico
ECCV	European Conference on Computer Vision
FN	Falsos Negativos
FP	Falsos Positivos
FPR	Tasa de Falsos Positivos (False Positive Rate)
FPS	Imágenes Por Segundo (Frames per second)
HMI	Interfaz Hombre Maquina (Human Machine Interface)
HOG	Histograma de Gradientes Orientados (Histogram of Oriented Gradients)
IDE	Entorno de Desarrollo Integrado (Integrated Development Enviroment)
IVVI	Vehículo Inteligente basado en Información Visual (Intelligent Vehicle based on Visual Information)
KNN	K Vecinos Próximos (K Nearest Neighbors)
LSI	Laboratorio de Sistemas Inteligentes
OMS	Organización Mundial de la Salud
PNB	Producto Nacional Bruto
ROC	Característica Operativa del Receptor (Receiver Operating Characteristic)
SDK	Kit de Desarrollo de Software (Software Development Kit)
SIFT	Escala Invariante de Características de la Transformada (Scale Invariant Feature Transform)
SPC	Especificidad
SURF	Transformación de la función Speeded-UP (Speeded-UP Feature transform)
SVM	Maquina de Soporte Vectorial (Support Vector Machine)
V2V	Vehículo a Vehículo
VN	Verdaderos Negativos
VP	Verdaderos Positivos

## **1. INTRODUCCIÓN**

La evolución tecnológica y el abaratamiento progresivo de los smartphones han hecho que su uso sea cada vez más habitual en el ámbito del desarrollo de aplicaciones profesionales que integren capacidades de alto coste computacional. Entre estas aplicaciones se encuentra el reconocimiento y la detección de objetos basado en el reconocimiento de imagen, posible de desarrollar con la tecnología actual a un coste relativamente bajo.

Por otro lado, el desarrollo continuo de los sistemas avanzados de asistencia a la conducción, (ADAS), introduce mecanismos que permiten mejorar la seguridad automovilística en el momento de la conducción con el propósito de minimizar los riesgos de sufrir un accidente vial o colisionar con otros vehículos. Esta tecnología se está integrando poco a poco en toda la industria automovilística y dará paso a afrontar nuevos retos dentro del campo, como por ejemplo el llamado coche conectado.

Este proyecto tiene como fin aunar la capacidad de procesamiento que ofrecen los smartphones actuales con los sistemas ADAS para dar lugar a una aplicación desarrollada para la plataforma Android que sea capaz de detectar ciclistas en la vía urbana durante la conducción del vehículo.

La detección de ciclistas se basa en dos etapas, una de generación de candidatos en la que se ha utilizado el descriptor HOG (Histograma de Gradientes Orientados) y otra etapa de clasificación de candidatos realizada mediante la utilización del clasificador entrenado mediante SVM (Máquina de Soporte Vectorial). Ambos métodos se encuentran presentes en las librerías OpenCV.

La aplicación como se ha mencionado anteriormente se ha desarrollado para funcionar sobre una plataforma Android, ya sea móvil o tableta. Dicha configuración hace que el sistema sea totalmente portable e integrable en todo tipo de vehículo terrestre además de hacerlo accesible para la gran mayoría de la población.

Al final del presente documento se reflejan los resultados obtenidos tras realizar una batería de pruebas a la aplicación final y se definen las posibles mejoras y ampliaciones a implementar en un futuro.

## 1.1 MOTIVACIÓN

El sector del transporte es un eje clave para el desarrollo económico y social del cualquier estado. Según los datos de Bussines Intelligence Insiders se estima que para finales de 2020 el número de vehículos terrestres completamente autónomos rondara la cifra de los 10.000.000 de unidades [1].

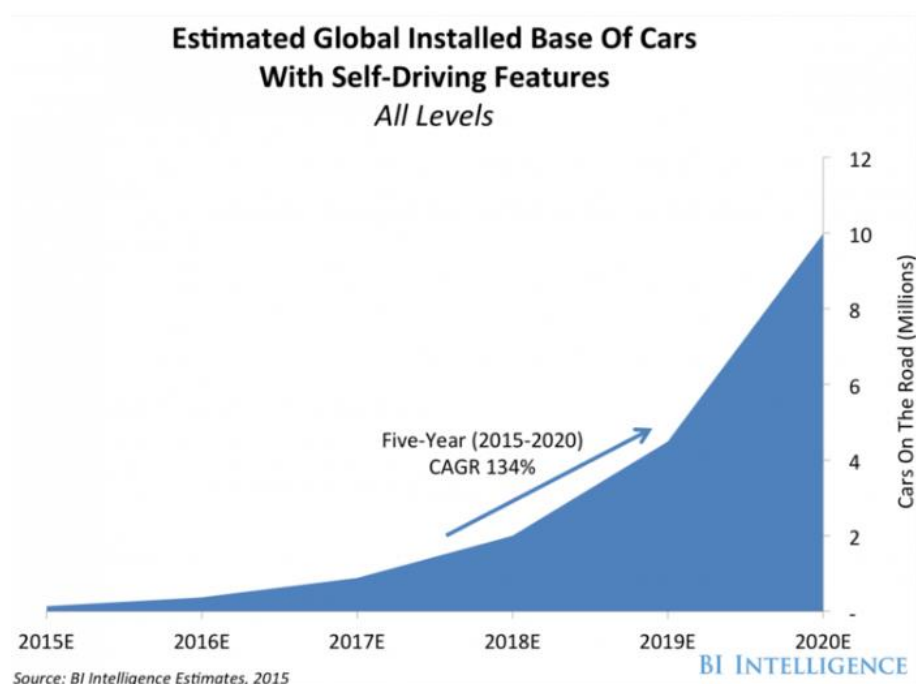


Ilustración 1-1 Estimación de crecimiento del parque automovilístico autónomo [1].

Las lesiones causadas por accidentes de tráfico constituyen un importante problema para la salud pública. Se estima que cada año en el mundo mueren 1,2 millones de personas a causa de colisiones en la vía pública y hasta 50 millones resultan heridas. Las proyecciones indican que, sin un renovado compromiso con la prevención, estas cifras aumentarán en torno al 65% en los próximos 20 años.

El Informe mundial sobre prevención de los traumatismos causados por el tráfico publicado conjuntamente por la Organización Mundial de la Salud (OMS) y el Banco Mundial [2], ha subrayado la preocupación internacional por el hecho de que los sistemas de tráfico inseguros estén dañando gravemente la salud pública y el desarrollo mundial. Considera que la cifra de lesiones causadas por el tráfico es inaceptable y en gran medida evitable y que una prevención eficaz y sostenible exige esfuerzos por parte de los gobiernos y la industria automovilística.

En España se producen anualmente al menos 90.000 accidentes de tráfico con víctimas, y cerca de 2.000.000 de accidentes con daños materiales [3]. Según datos de la Universidad de

Valencia entre el 70 y 90% de sendos accidentes se producen a causa del factor humano. Algunas de las medidas adoptadas por la Comisión Europea, en materia de seguridad vial para reducir la siniestralidad son

- Regulación de los límites de velocidad.
- Educación vial.
- Medidas para implementar dispositivos de seguridad activa en los vehículos.

Tipo de vía	Accidentes con víctimas		Fallecidos*	Heridos hospitalizados		Heridos no hospitalizados	
	Número	%	Número	Número	%	Número	%
Vías interurbanas	2.013	28%	48	322	49%	1.803	29%
Vías urbanas	5.173	72%	10	330	51%	4.470	71%
<b>Total</b>	<b>7.186</b>	<b>100%</b>	<b>58</b>	<b>652</b>	<b>100%</b>	<b>6.273</b>	<b>100%</b>

Ilustración 1-2 Datos accidentes de tráfico España 2015 [3]

Uno de los ejes fundamentales tratado en el informe Estrategia de Seguridad Vial 2010-2020 [4] elaborado por la DGT fue la seguridad en el vehículo. Según dicho informe se ha de mejorar el conocimiento del parque de vehículos rodante y sus condiciones de seguridad, se ha de promover la incorporación de los nuevos sistemas de seguridad activa y pasiva con el fin de reducir notablemente el número de accidentes y su gravedad. Algunas de las medidas propuestas por dicho informe son:

- Promocionar la demanda de elementos de seguridad activa y pasiva incorporada a los vehículos.
- Promover la renovación del parque de vehículos con criterios de seguridad vial.
- Desarrollar los sistemas de interacción vehículo-vehículo y vehículo-infraestructura.
- Realizar campañas para concienciar sobre la importancia del mantenimiento preventivo del vehículo desde la perspectiva de la seguridad vial.



Ilustración 1-3 Ejes de actuación Plan estratégico DGT [4]

Al problema de la siniestralidad causada por factores humanos hay que añadirle el impacto económico que conllevan los accidentes de tráfico. Estimar los costes de las colisiones y los traumatismos causados por el tráfico puede ayudar a los países a comprender la gravedad del problema que plantean y las ventajas de invertir en medidas para prevenirlos. La evaluación debería tener en cuenta los costes tanto directos como indirectos. Como mínimo, en los costes directos debería incluirse la prestación de atención sanitaria y rehabilitación, y en los indirectos el valor de los gastos ocasionados derivados de los cuidadores y las indemnizaciones para los supervivientes y las familias. A continuación, se muestran algunas cifras significativas sobre dichos costes:

- Los traumatismos causados por el tráfico cuestan a los países de la Unión Europea 180.000 millones de euros anuales, es decir, dos veces el presupuesto anual para todas las actividades en estos países [2].
- En los Estados Unidos, el costo anual asciende a 230.600 millones de dólares, es decir, el 2,3% del PNB (Producto Nacional Bruto) [2].
- En China, en 1999, los traumatismos causados por el tráfico causaron la pérdida de 12,6 millones de años de vida potencialmente productiva, con un valor estimado de 12.500 millones de dólares, casi cuatro veces el presupuesto anual del país en salud [2].
- La Fundación Instituto Tecnológico para la Seguridad del Automóvil calculó que, en España, en el año 2004, los accidentes de circulación supusieron para la sociedad en su conjunto un coste situado entre los 13.000 y los 17.600 millones de euros [5].

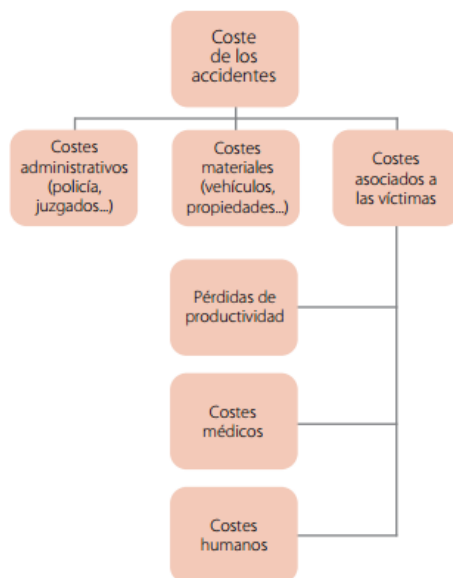


Ilustración 1-4 Desglose de costes por accidente de tráfico [5].

La inversión en desarrollo y el uso de tecnologías como los sistemas avanzados de asistencia a la conducción y la integración del coche autónomo en la sociedad harán posible en un futuro próximo una notable disminución del número de accidentes de tráfico y como consecuencia la reducción del número de víctimas y el coste que ambos conllevan. Además, el uso del coche autónomo traerá a la sociedad otro tipo de ventajas como el aumento de la eficiencia en la conducción, la reducción de la congestión del tráfico, el descenso de la contaminación atmosférica, la accesibilidad al coche autónomo a personas con discapacidad y ancianos, la desaparición de la responsabilidad civil de las personas físicas y la desaparición de las multas de tráfico.

En la actualidad el vehículo autónomo comienza a ganar fuerza. En agosto de 2016, Uber lanzó junto con Volvo un proyecto piloto en la ciudad de Pittsburgh, para que los vehículos autónomos comiencen a transportar usuarios. Por ahora se realiza con dos ingenieros de la compañía en los asientos delanteros, uno preparado para tomar los mandos del vehículo por si fuera necesario, y otro recabando datos sobre posibles incidencias. En Singapur, una startup ya ha comenzado a poner en circulación taxis autónomos y la ciudad de Helsinki ha puesto al servicio de los ciudadanos un autobús autónomo.

Grandes empresas llevan años invirtiendo en el desarrollo y mejora de sistemas ADAS como el control de crucero inteligente, sistema de parada automática en caso de emergencia, detección y aviso por cambio involuntario de carril, sistemas de detección de peatones y asistencia al aparcamiento entre otros.



Ilustración 1-5 Taxi y autobús autónomo.

Todos estos sistemas se desarrollan y utilizan en muchos casos como parte de la equipación de un vehículo en concreto y no es posible modificarlos una vez instalados. En conclusión, no existen soluciones ADAS en el mercado portables, fácilmente configurables e instalables.

De todas las tecnologías ADAS existentes, una de las más susceptibles de convertirse en un sistema portable es la detección de agentes de la vía mediante visión por computador. Esto es debido sobre todo al desarrollo de la tecnología para dispositivos móviles, ya que cámaras de pequeño tamaño y alta resolución son una realidad en el mercado actual

En el ámbito de los smartphones no existen apenas soluciones basadas en visión por computador ya que, hasta hace muy poco tiempo, estos dispositivos aún no tenían la capacidad de procesamiento suficiente como para ejecutar aplicaciones con un coste computacional alto. En la actualidad, el mercado ha cambiado ofreciéndonos productos que son capaces de cubrir las necesidades computacionales, posibilitando así el desarrollo del presente proyecto.

## 1.2 DESCRIPCIÓN DEL PROYECTO Y OBJETIVOS

Este proyecto consiste en el desarrollo de una aplicación implementada en Android para la detección y seguimiento de ciclistas permitiendo la carga de entrenamientos de forma fiable.

El objetivo principal de la aplicación es que el software desarrollado pueda ejecutarse en cualquier dispositivo portable de gama media o gama alta que corra bajo plataforma Android. Asimismo, tanto la funcionalidad para la detección como la de seguimiento utilizan entrenamientos basados en SVM (Maquinas de Vectores Soporte). Esta es una de las técnicas más poderosas en cuanto a entrenamiento se refiere, dotando a la aplicación junto con la detección por HOG de una gran sencillez en los modelos, gran robustez y buena generalización para nuevos datos.

Cabe mencionar que todos los entrenamientos SVM utilizados durante el desarrollo han sido proporcionados y elaborados por el Ingeniero Aurelio Ponz Vila, miembro del LSI (Laboratorio de Sistemas Inteligentes) de la Universidad Carlos III de Madrid.

Una de las funcionalidades principales implementadas en ambas aplicaciones es el algoritmo de carga y adaptación de los vectores SVM en un sistema Java y la preparación de los mismos para su posterior carga en las funciones de detección para HOG de las librerías OpenCv.

Para el algoritmo de seguimiento se ha utilizado el filtro de Kalman sumado al “Hungarian Method”, lo cual nos permite trazar una relación entre las antiguas entidades detectadas y los anteriores “tracks” en seguimiento. Este algoritmo ya se encontraba implementado en otras aplicaciones pertenecientes a la aplicación base del LSI y se ha adaptado para la correcta ejecución de las nuevas aplicaciones. Dentro de la aplicación de seguimiento cabe mencionar que también se han adaptado los métodos para la estimación de posición de las entidades en seguimiento.

Resumiendo, se ha realizado el desarrollo de dos aplicaciones, una para la detección y otra para el seguimiento y estimación de distancias para ciclistas. Ambas aplicaciones se han integrado en la aplicación base del LSI. El desarrollo en en Java permite la ejecución en dispositivos Android portables, posibilitando la detección y el seguimiento de ciclistas que circulen por la vía mediante la instalación del dispositivo en el parabrisas del coche. Aunando estas características se obtiene la implementación de un subsistema ADAS a un muy bajo coste con unas altas prestaciones.



## **2. ESTADO DEL ARTE**

Como ya se ha comentado en apartados anteriores, los sistemas ADAS convencionales permiten detectar objetos como peatones, vehículos, ciclistas o señales de tráfico; también permiten realizar operaciones de clasificación básicas, avisar al conductor si la carretera está en malas condiciones o entraña peligros y, en algunos casos, decelerar o detener el vehículo. El nivel de estos sistemas avanzados de ayuda a la conducción es excelente para el desarrollo de aplicaciones tales como el control de ángulos muertos, el cambio asistido de carril o el aviso de posibles colisiones.

En este apartado se tratará el estado de la técnica referente a aplicaciones desarrolladas para sistemas móviles que usen la tecnología de visión por computador orientadas a aumentar la seguridad vial. Por último, se desarrollarán conceptos importantes a conocer sobre los sistemas ADAS y las tecnologías aplicadas en proyectos académicos que tratan de cubrir la misma necesidad que este proyecto.

### **2.1 ADAS**

Los sistemas de asistencia a la conducción avanzada, de aquí en adelante ADAS por sus siglas en inglés, son sistemas que ayudan al conductor durante el trayecto. Los sistemas ADAS pretenden integrar el IHM en el vehículo con el propósito de aumentar en general la seguridad de conductor y pasajeros. Dichos sistemas son desarrollados para automatizar, adaptar y mejorar los sistemas del vehículo con el fin de brindar una mayor seguridad en la conducción.

Los equipos de seguridad están diseñados para evitar colisiones y accidentes, ofreciendo tecnologías que alertan al conductor de posibles problemas o simplemente para evitar colisiones mediante la advertencia al usuario en forma de alarmas o tomar el control del vehículo en caso necesario. Las funcionalidades adaptativas pueden automatizar la iluminación, proporcionar un control de cruce adaptativo, automatizar el proceso de frenado, incorporar advertencias GPS y de tráfico, mantener al conductor en el carril correcto, o mostrar entidades situadas en los puntos ciegos. Dichos sistemas basan su funcionamiento en las aportaciones de múltiples fuentes de datos, incluyendo captura de imágenes proporcionadas por sensores ópticos para su posterior procesamiento (visión artificial), la creación de redes inteligentes entre vehículos (V2V) etc.

En las próximas líneas se expondrán los subsistemas más importantes de los sistemas ADAS.

#### **2.1.1 ACC (Adaptive Cruise Control, control de velocidad adaptativo)**

El control de cruce adaptativo [6] amplía las funciones del sistema de control automático de la velocidad, añadiendo un sensor de radar para mantener automáticamente la distancia correcta con respecto al vehículo que circula por delante, dentro de ciertos límites del sistema.

Si el coche se aproxima a un vehículo que circula más lentamente por delante, el control de distancia automático decelera el coche hasta alcanzar su misma velocidad, a fin de mantener la distancia programada por el conductor. Si un vehículo que circula por delante se cambia al mismo carril, también se reduce la velocidad del coche para adaptarla a la del otro vehículo. Una vez despejada la carretera, el automóvil es acelerado hasta la velocidad de cruce seleccionada. De esta forma, el control de distancia automático mejora tanto el confort de conducción como la seguridad. Sin embargo, la responsabilidad última continúa recayendo en el conductor.

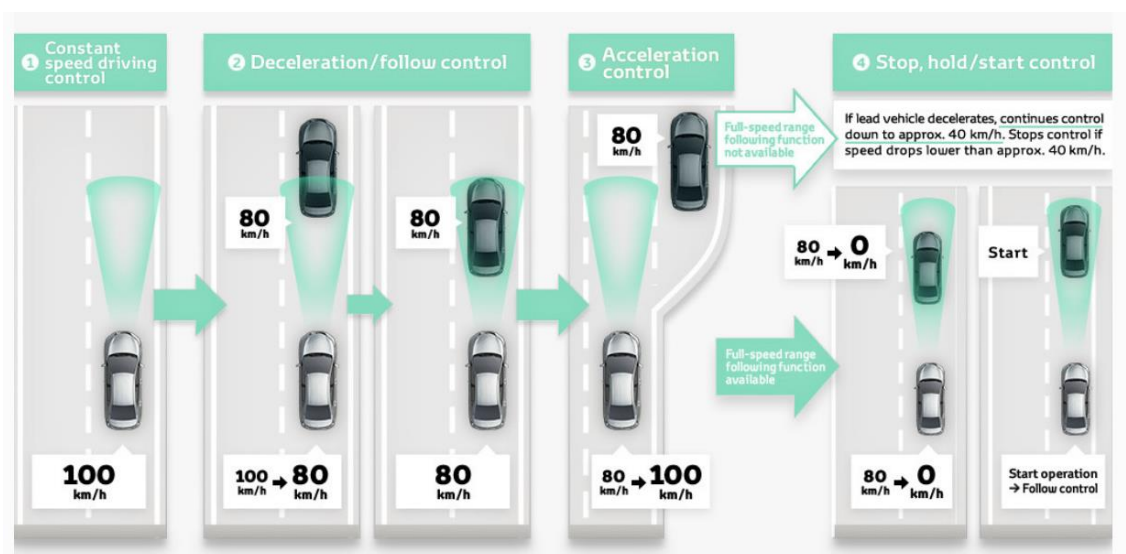


Ilustración 2-1 Ejemplo de funcionamiento de ACC desarrollado por Toyota.

Incluso con el control de distancia automático activado, el conductor sigue siendo responsable de controlar la velocidad del vehículo y la distancia con respecto al vehículo que circula por delante. En estos momentos el control de cruce adaptativo no reacciona a objetos fijos ni a vehículos que circulan en sentido contrario. El estado de la técnica actual no aconseja que este subsistema se utilice en carreteras tortuosas ni en condiciones meteorológicas adversas, tales como niebla, hielo o lluvia intensa.

Dentro de la tecnología de los ACC existen varias características que se encuentran disponibles en algunos sistemas entre las que se encuentran las siguientes:

### 2.1.2 Stop & Go

Stop & Go [7] o Asistente de conducción en atascos, consiste en la detección de vehículos a través de los sensores de radar y la cámara frontal para detectar el tráfico precedente. Con los datos procedentes de los sensores, el vehículo se mantiene en un carril virtual determinado por

el tráfico, de esta manera el vehículo es guiado automáticamente modificando su velocidad y dirección siempre y cuando se mantenga dentro de ciertos límites de velocidad. El uso de este tipo de tecnología tiene un impacto importante en la reducción de la emisión de gases contaminantes y la reducción de la contaminación acústica.



Ilustración 2-2 ACC. Stop & Go [8]

### **2.1.3 Asistente predictivo de eficiencia**

El Asistente Predictivo de Eficiencia [9] accede a los datos predictivos de la ruta del sistema de navegación (incluyendo límites de velocidad, pendientes y el radio de las curvas). La información del reconocimiento de la señalización de la carretera basado en cámaras también es usada para la detección señales de tráfico.

Si el control de crucero adaptativo está desactivado, el asistente de eficiencia predictivo solo tiene una función puramente informativa sin participación del control. Cuando el control de crucero adaptativo está activado, no sólo realiza indicaciones al conductor, sino que también interviene en el funcionamiento del vehículo, reduciendo la velocidad y actuando sobre las relaciones de cambio.

El conductor puede ignorar el control predictivo en cualquier momento pulsando el acelerador o el pedal del freno.



Ilustración 2-3 ACC -Asistente predictivo de eficiencia de Audi

#### **2.1.4 Asistente de esquiva o anticolidión**

En combinación con los sensores de radar de largo alcance del ACC y de cámara de alta resolución, el sistema anticolidión [10], [11] calcula una vía de escape adecuada usando la distancia, anchura y holgura transversal del vehículo precedente. El asistente anticolidión ayuda al vehículo a sortear obstáculos en situaciones críticas dentro de un intervalo funcional. También ayuda a corregir el movimiento de la dirección, siempre que el conductor esté maniobrando activamente.

#### **2.1.5 Asistente de Giro**

El asistente de giro [12] debe evitar que el vehículo colisione con otro vehículo que se acerque en sentido contrario, cuando se vaya a realizar un giro a la izquierda en un cruce, hasta una velocidad determinada usualmente baja, unos 10 km/h. Se controla continuamente el tráfico que se acerca en sentido contrario usando los radares del ACC y la cámara frontal. Al hacerlo, el sistema sabe cuándo un vehículo circula en sentido contrario. Simultáneamente, también se detecta que el conductor se propone girar a la izquierda por la activación del intermitente. En un primer momento los frenos se activan de manera automática para que el vehículo no cruce el carril y si las condiciones son óptimas se permite efectuar el giro.

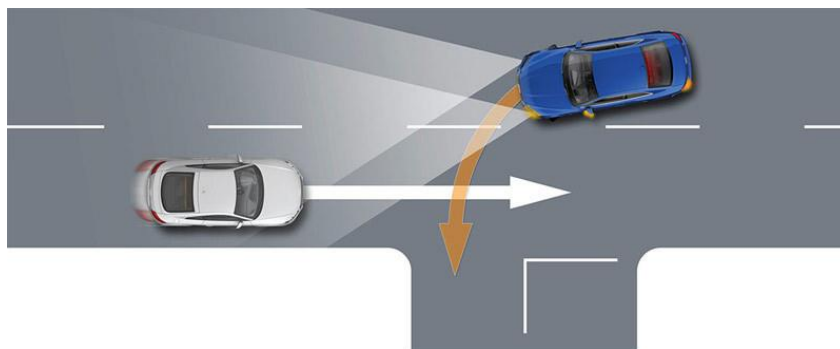


Ilustración 2-4 ACC -Ejemplo de situación del asistente de giro [13]

### 2.1.6 **AEB (Automatic Emergency Braking, freno de emergencia automático)**

El frenado automático [14] es una tecnología para detectar una inminente colisión con otro vehículo, persona, o cualquier otro tipo de obstáculo, respondiendo mediante la aplicación de los frenos para detener el vehículo sin intervención del conductor. Dependiendo del fabricante puede disponer de sensores, radar, vídeo, tecnologías de infrarrojos, ultrasonidos, etc.

Estos dispositivos pueden detectar peligros fijos, tales como postes o señales verticales de tráfico, a través de una base de datos de localización. En el caso de intervenir el conductor para evitar la colisión mediante alguna maniobra como, por ejemplo, pisar el pedal del freno o mover la dirección, el dispositivo se desactiva.

El sistema de frenado automático está pensado para que, en entornos urbanos, una distracción no termine en colisión o accidente por alcance. Básicamente el sistema está atento al entorno inmediato, es decir, a una distancia de seguridad medida por delante del vehículo y si se detecta una aproximación a un obstáculo a una velocidad alarmante el sistema frena evitando la colisión. Si tras la frenada no se pisa el embrague, el coche se cala e inmediatamente activa los indicadores de emergencia.



Ilustración 2-5 ACC -Ejemplo de actuación [15]

## 2.2 APLICACIONES MOVILES

En las siguientes líneas se hará un repaso de las principales aplicaciones ADAS disponibles para sistema operativo Android que intentan cubrir necesidades en materia de seguridad vial.

### 2.2.1 Aplicaciones

#### 2.2.1.1 Movon FCW

Movon FCW [16] es una aplicación disponible en Google Play de manera gratuita, que usando la cámara de visión trasera del dispositivo móvil advierte al conductor de peligro por colisión con otros vehículos que circulen por la vía.



Ilustración 2-6 Aplicación MOVON FCW [17]

Entre las principales características de la aplicación se encuentran:

- Crear un sistema inteligente que reconoce vehículos mediante el uso de la cámara del dispositivo móvil.
- Muestra al usuario la distancia entre los dos vehículos y el tiempo para colisión.
- Sistema de alertas sonoras y visuales que advierten al conductor de un peligro por colisión inminente.

#### 2.2.1.2 AcoDriver5

AcoDriver 5 es una aplicación móvil disponible para sistemas Android. La aplicación hace las funciones de copiloto ofreciendo al usuario una experiencia de ayuda a la conducción haciendo uso de los recursos del smartphone tales como la cámara trasera para realizar la captura de video y el receptor GPS para realizar los cálculos de velocidad del automóvil.



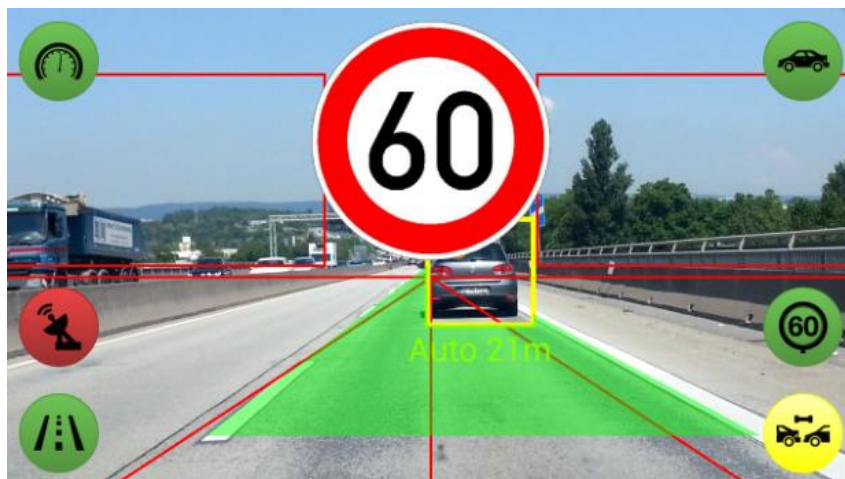


Ilustración 2-7 Aplicación AcoDriver 5 [18]

Las principales características de AcoDriver son:

- Detección y supervisión de los límites de velocidad en base al procesado de imágenes obtenidas mediante la cámara del dispositivo móvil.
- Control de la distancia entre el vehículo del usuario y el siguiente vehículo situado en el mismo carril de la vía.
- Control de posicionamiento en carril y uso de alarmas sonoras si el vehículo sobrepasa una línea continua de tránsito.

### 2.2.1.3 iOnRoad Augmented Driving 3

iOnRoad aprovecha los recursos de los smartphones avanzados para optimizar la conducción en tiempo real. La app se sirve de la cámara y los sensores GPS incorporados en el Smartphone para detectar cualquier vehículo que tengamos delante y alertar al conductor de posibles peligros. El radar visual informa en tiempo real de cualquier objeto que se halle frente al conductor y calcula la velocidad actual del vehículo basándose en los sensores incorporados en el dispositivo. Cuando el vehículo se aproxima a un peligro se dispara una señal audiovisual que alerta de una posible colisión, lo que permite al conductor frenar a tiempo.

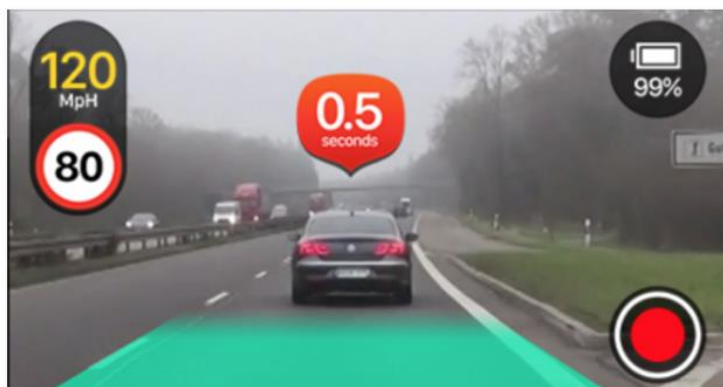


Ilustración 2-8 Aplicación AcoDriver 5 [19].

Como características principales posee:

- Advertencia de colisión frontal.
- Advertencia de abandono de carril.
- Advertencia de distancia de seguridad.
- Localizador del vehículo.
- Reconocimiento de señales de tráfico, alertando al conductor de exceso de velocidad.
- Registro completo de conducción e incidencias de riesgo.

#### 2.2.1.4 Mobile eye 2

Mobile eye basa su funcionamiento en la captura de imágenes mediante la cámara del Smartphone para el reconocimiento de objetos y el sistema de posicionamiento GPS para realizar el cálculo de la velocidad del vehículo.



Ilustración 2-9 Aplicación Mobile Eye 2.



El sistema ofrece al usuario las siguientes prestaciones:

- Sistema de advertencia de colisión: alerta sobre una posible colisión contra el vehículo le precede, mostrando un ícono con un automóvil rojo.
- Indicación de vehículo lejos: se muestra un ícono con un automóvil verde para indicar que el vehículo que va adelante se encuentra a una distancia prudencial del vehículo propio.
- Advertencia de Colisión con Peatones: alerta sobre una posible colisión con un peatón o ciclista se encuentre en frente del vehículo. Se muestra un ícono con un peatón rojo.
- Sistema de reconocimiento de señales de velocidad: El sistema reconoce mediante el procesado de imágenes las señales de velocidad, advirtiéndole al usuario si está es sobrepasada en algún momento.

### **2.2.2 Conclusiones**

El desarrollo de aplicaciones Android para smartphones en materia de seguridad vial es en estos momentos relativamente escaso. Las aplicaciones existentes son muy parecidas en cuanto a funcionalidad cumpliendo requisitos de detección de señales de tráfico, detección de vehículos y advertencias por abandono de carril. En cuanto a la detección de ciclista que es el objetivo de este proyecto, solamente se encuentra implementada en mobileye. Sin embargo, esta aplicación, aunque si avisa al usuario mediante una alarma sonora de la presencia de un ciclista no es capaz de situar al objeto dentro de la composición de la imagen y dicha detección está más orientada a circulación urbana a bajas velocidades del vehículo.

## **2.3 TÉCNICAS PARA LA DETECCIÓN Y CLASIFICACIÓN**

La detección de ciclistas en secuencias reales resulta ser un reto debido a las variadas formas en las que se pueden encontrar los mismos. A continuación, se expondrán las diferentes técnicas para su detección y clasificación englobadas en lo que se conoce como visión por computador, que consiste en emplear ordenadores para inferir la información dentro de imágenes digitales o digitalizadas.

### **2.3.1 Técnicas de extracción de características**

El proceso de extracción de características dentro de una imagen digital nos permite identificar objetos dentro de la misma, En ocasiones viene precedido por un preprocesado de la señal, maniobra necesaria para corregir posibles deficiencias en los datos debido a errores del sensor, o bien para preparar los datos de cara a posteriores procesos en las etapas de

extracción de características o clasificación.

Las características elementales están explícitamente presentes en los datos adquiridos y pueden ser procesados directamente en la etapa de clasificación. Las características de alto orden son derivadas de las elementales y se generan debido a manipulaciones o transformaciones en los datos.

A continuación, se detalla brevemente el funcionamiento de algunas de las técnicas más relevantes:

#### **2.3.1.1 Descriptores tipo de Haar-like**

Este método fue propuesto por Viola y Jones en 2004 [20]. Los descriptores Haar-like permiten definir de manera robusta clases de objetos complejos, siendo invariantes a cambios de color y de textura. Se emplean habitualmente para la detección de personas. Presentan la capacidad de codificar rasgos tales como cambios de intensidad en diferentes escalas. En concreto, los descriptores tipo Haar-like consisten en que la imagen es recorrida mediante una ventana a la que se le aplican varios clasificadores en serie, cada uno más complejo que el anterior, los cuales usan las características para confirmar o descartar la hipótesis de que se trata del objeto buscado. Si la hipótesis se rechaza en cualquier nivel, el proceso no continúa, pero si se confirma significará que se ha detectado el objeto deseado. Los patrones se evalúan sobre distintos ángulos. Además, el algoritmo puede ejecutarse a varias escalas para identificar objetos de diferentes tamaños o de tamaño desconocido.

#### **2.3.1.2 SIFT (Scale-invariant feature transform)**

SIFT es un método propuesto por David Lowe en 1999 [21], que se centra en busca de puntos característicos que cumplen criterios espacio-escalares. Los descriptores se calculan a través de la orientación de los gradientes de cada punto. Una vez extraídos los puntos característicos invariantes y distintivos de una imagen pueden ser usados para mejorar la correspondencia entre dos vistas diferentes de un objeto o una escena.

#### **2.3.1.3 SURF (Speeded Up Robust Feature)**

Uno de los sucesores más importantes de SIFT, ha sido el algoritmo Speeded-Up Robust Features (SURF) [22]. SURF fue presentado en 2006 en el ECCV en Graz (Austria). Está parcialmente inspirado en SIFT y se ha demostrado que en prácticamente todos los casos consigue mejorar el rendimiento de este algoritmo. Está basado en el cálculo del determinante de la matriz Hessiana (DoH: Determinant of Hessian) para la detección de puntos interesantes y en las wavelets de Haar para la descripción de dichos puntos. Esta aproximación es aún más

rápida que DoG (la utilizada por SIFT) y ofrece una respuesta superior en cuanto a calidad de detección de las imágenes.

#### 2.3.1.4 HOG

Este método [23] fue presentado por Navneet Dalal y Bill Triggs en el Instituto Nacional de Investigación en Informática y Automática (INRIA), en 2005. Consiste en la división de la imagen en subbloques distribuidos a lo largo y ancho de la misma con cierto grado de solapamiento entre ellos. Cada bloque se subdivide en subbloques (o celdas) y sobre estos últimos se calcula la magnitud y orientación de los gradientes en cada píxel. Sobre cada uno de estos bloques se calcula el histograma de los gradientes orientados promediado por un peso gaussiano, y luego se almacena en el vector de características de la imagen.

El descriptor se compone de celdas  $M * N$  que dividen la imagen en una cuadrícula. Cada celda está representada por un histograma de orientaciones de borde, donde el número de orientaciones de bordes es un parámetro (usualmente 9). El histograma de celdas se visualiza mediante una "estrella" que muestra la fuerza de las orientaciones de los bordes en el histograma: cuanto más fuerte es una orientación específica, más larga es respecto a las otras.

Obsérvese que existen varios esquemas de normalización: esquemas locales, en los que la celda se normaliza con respecto a las celdas vecinas solamente (como en el documento original de Dalal-Triggs), o esquemas globales, en los que la longitud de orientación es normalizada por todas las celdas.

El ejemplo siguiente muestra un modelo de una bicicleta con HOG que consta de  $7 * 11$  celdas, cada una con 8 orientaciones.

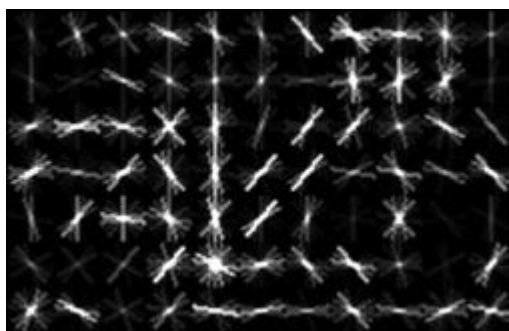


Ilustración 2-10 HOG aplicado a imagen de bicicleta

HOG es muy efectivo para identificar objetos con una figura muy marcada, como pueden ser seres humanos. En este trabajo hemos empleado los descriptores HOG, dado su robustez frente a diferentes condiciones de iluminación, pequeños cambios en el contorno de la imagen, diferentes fondos y escalas, y a la ya nombrada efectividad con objetos con una marcada figura.

### **2.3.2 Clasificadores**

Los clasificadores son algoritmos capaces de aprender una cierta distribución de datos a partir de una serie de ejemplos de entrenamiento, para posteriormente poder predecir la clase a la que pertenecen nuevos ejemplos no utilizados en el entrenamiento. Podemos encontrar varias ramas de clasificadores según su aprendizaje:

#### **2.3.2.1 Aprendizaje no supervisado**

En este caso, el conjunto de entrenamiento no dispone de etiquetas conocidas, así que requiere de técnicas de agrupamiento que intenten construir estas etiquetas. El aprendizaje semi-supervisado es una combinación del aprendizaje supervisado y del no supervisado. Surge de la dificultad que conlleva obtener los datos etiquetados requeridos en el aprendizaje supervisado. Por esa razón este método recurre al uso de una parte de datos etiquetados y un conjunto más extenso sin etiquetar mejorando, de este modo, la construcción de los modelos. Se asume que los datos no etiquetados siguen la misma distribución que los etiquetados

#### **2.3.2.2 Aprendizaje por refuerzo**

Este método no trata de aprender a partir de un conjunto de ejemplos si no a través de la experiencia.

#### **2.3.2.3 Aprendizaje supervisado**

La detección se puede realizar mediante el aprendizaje automático de las diferentes vistas de un objeto extraídas de un conjunto de ejemplos empleados como plantillas, es decir, a partir de un set de ejemplos de aprendizaje se genera una función que relaciona las entradas con las salidas de interés. Se considera muy importante la correcta selección de las características o descriptores de los objetos que se van a tomar para la clasificación. Uno de los inconvenientes de este método es que, para obtener resultados precisos, se requiere de un gran conjunto de ejemplos para la realización del aprendizaje. A día de hoy son las técnicas de aprendizaje más comunes, debido a su sencillez de implementación, y a su robustez. Cabe destacar las siguientes técnicas dentro de este ámbito:

#### 2.3.2.3.1 ADAPTIVE BOOSTING

Las técnicas de Boosting son métodos iterativos que obtienen clasificadores muy precisos mediante la combinación de muchos clasificadores base no tan precisos. Estos clasificadores base se distribuyen en grupos y, a su vez, estos grupos denominados “etapas” se enlazan formando una cascada y, actuando cada uno sobre las predicciones del anterior, dan lugar al clasificador final (Paul Viola y Michael Jones) [20].

#### 2.3.2.3.2 ALGORITMO KNN

El algoritmo KNN (K Nearest Neighbors) [24] [25] es un método de clasificación supervisada que sirve para estimar la probabilidad de que un elemento  $x$  pertenezca a la clase  $C(j)$  a partir de la información proporcionada por el conjunto de prototipos, donde  $k$  determina el número de vecinos que son contemplados para realizar la clasificación. En el reconocimiento de patrones, el algoritmo KNN es usado como método de clasificación de objetos basado en un entrenamiento mediante ejemplos cercanos en el espacio de los elementos.

#### 2.3.2.3.3 SUPPORT VECTOR MACHINES (SVM)

Las Máquinas de Soporte Vectorial (SVM) [26] son un conjunto de algoritmos de aprendizaje supervisado empleados para la clasificación y la regresión desarrollados por Vladimir Vapnik. Dado un conjunto de ejemplos de entrenamiento (muestras) podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Tomando los datos de entrada como conjuntos de vectores en un espacio  $n$ -dimensional, una máquina de vectores soporte construirá un hiperplano de separación en ese espacio. Se considera que es mejor clasificador de datos aquel hiperplano que maximice la distancia (o margen) con los puntos que estén más cerca de él. Siendo los vectores de soporte los puntos que tocan el límite del margen.

Debido a que es el método empleado en este proyecto, se ha considerado oportuno dar una explicación más detallada de su funcionamiento que ha sido incluida en la sección 4.5.1.

### 3. DESCRIPCIÓN GENERAL DEL PROYECTO

#### 3.1 CONTEXTO

En las siguientes líneas se expondrán el conjunto de circunstancias que rodean al desarrollo de la aplicación para la detección de ciclistas. Dicha aplicación forma parte como ya se ha citado anteriormente, de la aplicación base del LSI. La aplicación la elaboró Alejandro Ramos López como Trabajo de Fin de Grado [27] y se ha ido completando con diversas aplicaciones elaboradas por otros alumnos con el fin de ampliar la funcionalidad y capacidades de la misma.

Actualmente la aplicación base cuenta con las siguientes capacidades:

- Detector de Peatones.
- Detector de Coches.
- Detector de Ciclistas.
- Ejemplo Libro OpenCV.
- Seguimiento de Peatones.
- Seguimiento de Coches.
- Seguimiento de Peatones XY.
- Seguimiento de Coches XY.
- Seguimiento de Ciclistas.



Ilustración 3-1 App Base LSI.

El objetivo final es que una vez desarrolladas todas las funcionalidades y aunadas todas las capacidades, la aplicación base sirva como parte del proyecto IVVI 2.0 del LSI para dotarle de las características ADAS que recoge la aplicación.

### 3.1.1 LSI UC3M

El laboratorio de sistemas inteligentes de la Universidad Carlos Tercero de Madrid ha desarrollado IVVI 2.0 [28], un vehículo provisto de un sistema de ayuda a la conducción que es capaz de detectar peatones en la vía gracias a su sistema de cámaras infrarrojas. Dicha tecnología se nutre de la detección del calor corporal humano como dato y logra localizar personas en condiciones de muy baja o nula visibilidad. El sistema es capaz de detectar peatones a 40 metros. IVVI tiene otras muchas capacidades como la detección de las líneas viales, detección y reconocimiento de señales de tráfico o vigilancia del rostro del conductor y alerta del mismo en caso de distracción o somnolencia.



Ilustración 3-2 IVVI uc3m

## 3.2 PROPOSITO Y CARACTERÍSTICAS GENERALES DEL PROYECTO

El propósito del proyecto es ampliar la funcionalidad de la aplicación base del laboratorio de sistemas inteligentes y que dicha funcionalidad permita añadir clasificadores de forma fiable de modo que una vez cargados dichos entrenamientos el sistema realice detecciones, en este caso de ciclistas desde cualquier perspectiva y a posteriori realizar un seguimiento óptimo de los mismos. Uno de los requisitos que se debían cumplir dentro del proyecto es que la detección se debía de realizar mediante el uso del detector HOG y un clasificador SVM.



Con este propósito el proyecto incluye:

- La evaluación y resultados de tres clasificadores SVM de ciclistas.
- El desarrollo de una actividad Android dentro de la aplicación para realizar la detección de ciclistas.
- Una actividad Android para realizar el seguimiento de los ciclistas.
- el desarrollo de las funciones pertinentes para poder cargar e implementar fácilmente cualquier clasificador producto de un entrenamiento de manera fiable.

### 3.3 DESCRIPCIÓN DE HERRAMIENTAS HARDWARE Y SOFTWARE.

A continuación, se realiza una descripción de las herramientas Hardware y Software que han sido necesarias para la ejecución de proyecto.

#### 3.3.1 HARDWARE

Durante la ejecución del proyecto se han realizado las pruebas con dos dispositivos móviles de la marca Xiaomi. El primero de los dispositivos es el Xiaomi Redmi 3, un móvil de gama media-baja y de bajo coste cuyas especificaciones y apariencia se detallan a continuación.



Ilustración 3-3 HW Xiamo Redmi 3

Características relevantes:

- Modelo Procesador: Qualcomm Snapdragon 616 MSM8939v2
- CPU: 4x1.7 GHz ARM Cortex-A53+ 4x1.2 GHz ARM Cortex-A53
- Tipo Procesador: Octa-Core
- Frecuencia de Reloj: 1.7 Ghz.



- Arquitectura: 64 Bits
- GPU: Qualcomm Adreno 405.
- Memoria Ram: 2 GB.
- Test Antutu: 33.726 (Antutu v6) Rendimiento mejor que el 61% de los dispositivos.
- Precio: 122 €.

El otro dispositivo móvil es el modelo Mi5 Advanced de Xiaomi, que, aunque tiene un coste reducido se podría categorizar como un móvil de altas prestaciones. A continuación, se detallan algunas de sus características.



Ilustración 3-4 HW Xiaomi MI5

Características relevantes:

- Modelo Procesador: Qualcomm Snapdragon 820 MSM8996 2.15 GHz.
- CPU: Dual cluster Kryo 2x2.15GHz + 2x1.59GHz.
- Tipo Procesador: Quad-Core.
- Frecuencia de Reloj: 2.15 GHz.
- Arquitectura: 64 Bits
- GPU: Adreno 530 624MHz.
- Memoria Ram: 3 GB.
- Test Antutu 136.773 (Antutu v6) Rendimiento mejor que el 96% de los dispositivos.
- Precio: 198 €.

El motivo de incluir el Xiaomi mi5 en el Hardware del proyecto fue principalmente que, durante

la ejecución de las pruebas, se observó que el Xiaomi redmi 3 aunque manifestaba un rendimiento óptimo durante la ejecución de las mismas, al cabo de un tiempo se sobrecalentaba y provocaba el apagado de seguridad del dispositivo.

El equipo utilizado para el desarrollo de la aplicación es un PC de sobremesa clónico con las siguientes características:

- CPU: AMD Phenom II x4 965
- Tipo Procesador: Quad-Core.
- Frecuencia de Reloj: 3.4 Ghz.
- Arquitectura: 64 bits.
- Memoria Ram: 16 GB.
- Tipo de disco duro: Estado Sólido.

### **3.3.2 SOFTWARE**

A continuación, se realizará una breve exposición de las herramientas Software utilizadas durante la ejecución del proyecto

### **3.3.3 Librerías OpenCv**

OpenCV (Open Source Computer Vision Library) [29] es una biblioteca de software de visión artificial y aprendizaje de máquina de código abierto multiplataforma pudiendo ser utilizado con C ++, C, Python, Java y MATLAB soportando Windows, Linux, Android y Mac OS. Estas librerías fueron creadas para proporcionar una infraestructura común para aplicaciones de visión por computador y para acelerar el uso de la percepción de máquinas en productos comerciales. Al ser un producto con licencia de BSD, OpenCV facilita que las empresas utilicen y modifiquen el código.

OpenCV ha sido diseñado para ser eficiente en cuanto a gasto de recursos computacionales y con un enfoque hacia las aplicaciones de tiempo real. La biblioteca cuenta con más de 2500 algoritmos optimizados, que incluye un amplio conjunto de utilidades de última generación de visión por computador y algoritmos de aprendizaje. Estos algoritmos pueden ser utilizados para detectar y reconocer rostros, identificar objetos, clasificar acciones humanas en videos, realizar un seguimiento de movimientos de cámara, rastrear objetos en movimiento, extraer modelos 3D de objetos, producir nubes de puntos 3D desde cámaras estéreo, unir imágenes juntas para producir una alta resolución Imagen de una escena entera, buscar imágenes similares de una base de datos de imágenes, eliminar los ojos rojos de las imágenes tomadas con flash, seguir los movimientos de los ojos, reconocer el paisaje y establecer marcadores para superponerlo con la realidad aumentada. OpenCV tiene más de 47 mil personas de usuario comunidad y número estimado de descargas superiores a 14 millones. La biblioteca se utiliza ampliamente

en empresas, grupos de investigación y organismos gubernamentales.

### 3.3.4 Android Studio

Android Studio [30] fue anunciado en el año 2013 en la conferencia de Google I/O. Fue creado para reemplazar a Eclipse, la plataforma que se usaba para la creación de aplicaciones y que todavía hoy en día es utilizada por muchos programadores. De esta manera, con Android Studio, Google consigue su propio IDE para el desarrollo de aplicaciones, pudiendo instalar todo el SDK para desarrollar apps específicas adaptadas a la mayor parte de versiones.

En la actualidad Android Studio es la plataforma que se postula como el más completo IDE para desarrollar aplicaciones Android con muchas características que destacan sobre los otros programas usados para el mismo ámbito. Está basado en IntelliJ y puede ser descargado de forma gratuita a través de la licencia de Apache 2.0.

Se ha escogido esta herramienta como plataforma de trabajo por la sencillez de su estructura a la hora de organizar el proyecto y por poseer un entorno fácil e intuitivo ya que permite desarrollar y ver en tiempo real las aplicaciones y las pantallas en las que será usada la aplicación.

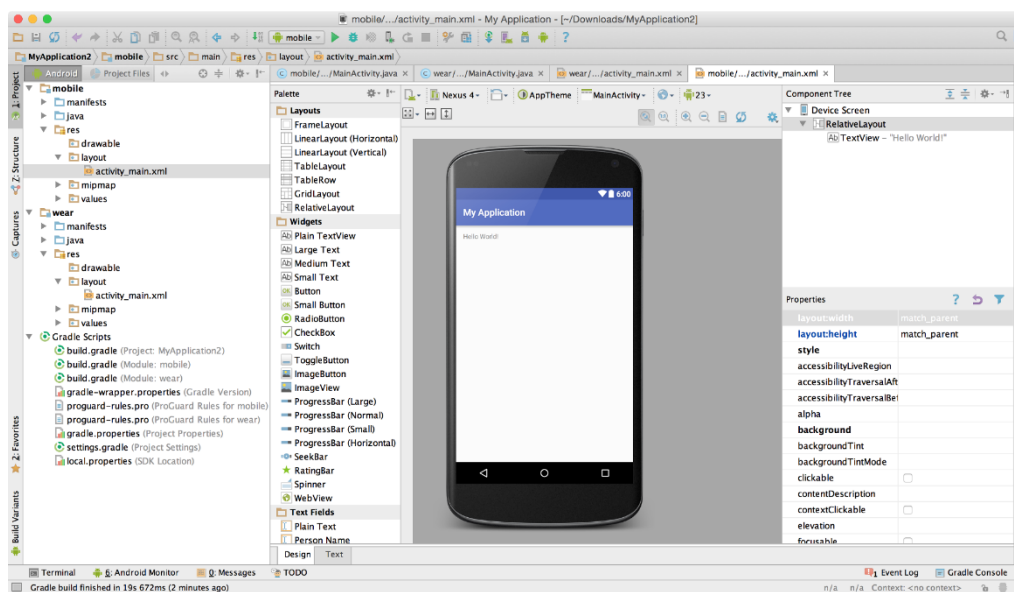


Ilustración 3-5 Android Studio

Android Studio v 2.2 ha sido la IDE elegida para el desarrollo de la aplicación junto con la versión 3.0 de las librerías OpenCV para Android.

### 3.3.5 **SVMLight**

SVMLight es una implementación de Máquina de Soporte Vectorial para solucionar los problemas de reconocimiento de patrones, el problema de regresión, y el problema de aprendizaje de funciones de clasificación. Los algoritmos de optimización utilizados en SVMLight se describen en [31]. El algoritmo tiene requisitos de memoria escalables y puede solucionar miles de problemas para SVM de manera eficiente.

Las principales características del programa son las siguientes:

- Algoritmo de optimización rápida
- Resuelve problemas de clasificación y regresión.
- Resuelve problemas de clasificación.
- Calcula las estimaciones de  $\text{XiAlpha}$  de la tasa de error.
- Calcula de manera eficiente las estimaciones de "Leave-One-Out" de la tasa de error y precisión.
- Incluye algoritmos para el entrenamiento aproximado de SVMs transductores grandes (TSVMs) (véase también Spectral Graph Transducer).
- Puede capacitar SVMs con modelos de costes.
- Maneja miles de vectores de soporte
- Maneja cientos de miles de ejemplos de entrenamiento
- Soporta funciones estándar del kernel y permite su definición.

SVMLight ha sido utilizado por el LSI de la universidad para realizar los entrenamientos usados en este proyecto.

## **4. DESCRIPCIÓN DETALLADA DE FUNCIONAMIENTO**

### **4.1 ASPECTOS BÁSICOS DE ANDROID**

Como ya sabemos, Android es un Sistema Operativo basado en el kernel de Linux y creado para su adaptación a sistemas portables con pantalla táctil. Google liberó la mayor parte del código de Android bajo licencia Apache, libre y de código abierto.

La estructura del sistema está formada por aplicaciones que se ejecutan dentro de un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Actualmente (a partir de Lollipop), Google anunció so comenzaría a utilizar JAVA ARM que incluye herramientas para desarrollo sobre ARM.

Dentro de la aplicación base del LSI, hasta ahora nos hemos referido al nuevo desarrollo como dos aplicaciones integradas que a partir de ahora y en este contexto pasamos a referirnos a ellas como “Activities” o actividades.

Una Activity es un componente de la aplicación que contiene una pantalla con la que los usuarios pueden interactuar para realizar una acción. Para cada una de las actividades se le puede asignar una ventana que a su vez hace de interfaz de usuario. En base a eso, una aplicación Android no es más que un conjunto de actividades que interactúan unas con otras ya que cada actividad puede a su vez iniciar otra actividad para poder así realizar diversas acciones.

Cada vez que se inicia una nueva actividad, la anterior es detenida pero el sistema la conserva en la pila de actividades. Ahora la actividad nueva posee el foco del usuario. La pila de actividades funciona como una LIFO por lo que cuando un usuario pulsa el botón “Atrás” la actividad con el foco se destruye y se recupera la actividad anterior. Cuando es detenida una actividad porque otra ha sido iniciada, se notifica el cambio de estado a través de los métodos “Callback”.

Para el desarrollo de una aplicación en Android es fundamental conocer este ciclo de vida de las actividades ya que se fundamentan en una serie de métodos que se pasan a detallar a continuación.



detalladamente en los siguientes apartados del presente capítulo.

Para finalizar este apartado, reseñar que cada una de las actividades desarrolladas tienen dos partes bien diferenciadas, la primera sería el layout o la configuración de elementos que van a formar parte de la HMI (Human Machine Interface) de la aplicación, y la otra parte haría referencia al conjunto de métodos y algoritmos que van a dar funcionalidad y uso a la aplicación. Ambas partes quedarán detalladas en los sucesivos apartados.

## 4.2 HUMAN MACHINE INTERFACE Y LAYOUT

El Layout es un objeto que contiene todas las vistas que posee una actividad. En el Layout se definen toda la estructura y elementos para que el usuario pueda interactuar con la aplicación por medio de la interfaz gráfica.

Para definir un Layout se define un documento XML que Android Studio genera automáticamente gracias a su panel de diseño.

En el caso de las dos actividades desarrolladas el Layout de la aplicación va a heredar de la clase `JavaCameraView` perteneciente a la librería OpenCV. Esta clase hace la función de nexo entre la librería OpenCv y la cámara del dispositivo.

A efectos prácticos el IHM de las actividades consta de una vista de las imágenes que está procesando la aplicación procedente de la cámara del móvil con una barra superior que muestra en la parte izquierda el logotipo del LSI y en la derecha un desplegable de opciones para configurar la aplicación. Por medio del botón “Atrás” del dispositivo se destruirá la actividad y se volverá al menú base de la actividad en ejecución.

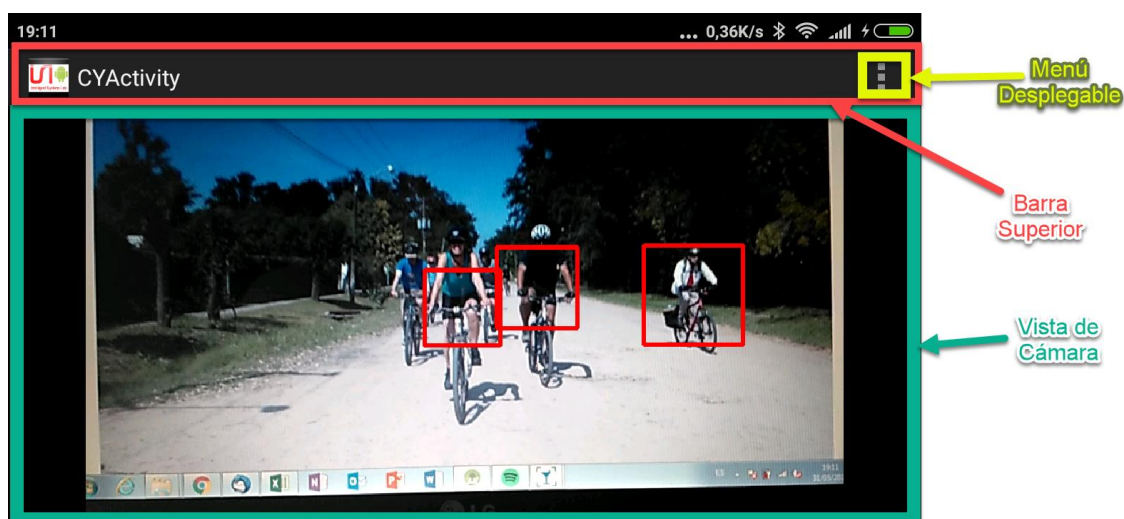


Ilustración 4-2 Aspecto de la aplicación desarrollada

Para ambos Layout se ha forzado que la vista de ejecución de la actividad siempre se realice en configuración “Landscape” ya que ejecutar la aplicación en vertical ocasionaría una pérdida de visión periférica de la imagen a procesar y podría incurrir en un accidente en un uso real de la misma.

### 4.3 ASPECTOS BÁSICOS DE LA APLICACIÓN

### 4.4 OPCIONES DE LA APLICACIÓN

A continuación, se detallan las opciones de configuración disponibles para cada una de las aplicaciones:

- **Detector de Ciclistas:**
  - **Cambiar Resolución.** Despliega todas las resoluciones soportadas por el dispositivo. Para las pruebas se ha utilizado una resolución de 384x288 pixeles. A mayor resolución aumenta el procesamiento y el número de FPS que el sistema es capaz de procesar. -
- **Seguimiento de Ciclistas:**
  - **Cambiar Resolución.** Despliega todas las resoluciones soportadas por el dispositivo. Para las pruebas se ha utilizado una resolución de 384x288 pixeles. A mayor resolución aumenta el procesamiento y el número de FPS que el sistema es capaz de procesar. -
  - **Modificar Margen.** Opción para reducir la ventana de detección en un porcentaje por la parte superior. Esta opción se utiliza para mejorar el rendimiento de la aplicación reduciendo la detección por el margen superior por el cuál no se van a detectar ciclistas ya que estos siempre aparecerán a ciertas alturas y a lo largo del eje horizontal.
  - **Seleccionar Altura.** Dato importante que introducir antes de empezar a utilizar la aplicación. Se refiere a la altura a la que se encuentra el dispositivo del suelo y la aplicación lo utiliza en el algoritmo de estimación de distancias.
  - **Registrar Datos.** Utilidad para el registro de datos usada para la evaluación de resultados. Vuelca datos relativos a la estimación de distancias y tracks a un fichero de texto.
  - **Capturar Frames.** Utilidad para el registro de datos usada para la evaluación de resultados. Realiza una captura de todos los frames que han sido procesados.



## 4.5 ALGORITMIA PARA LA DETECCIÓN

La detección como ya se ha introducido anteriormente se ha realizado utilizando entrenamientos obtenidos mediante clasificador SVM para luego realizar la detección mediante HOG. A continuación, se profundizará en estos dos métodos y se detallará como se han implementado en la aplicación.

### 4.5.1 Support Vector Machine.

A la hora de realizar la clasificación se ha empleado un algoritmo de SVM basado en características HOG, a continuación, se detalla de forma genérica las bases del método de clasificación SVM:

Las SVMs tienen su origen en los trabajos sobre la teoría del aprendizaje estadístico y fueron introducidas en los años 90 por Vapnik et al. [32]. Aunque originariamente las SVMs fueron pensadas para resolver problemas de clasificación binaria, actualmente se utilizan para resolver otros tipos de problemas (regresión, agrupamiento, multclasificación).

Dentro de la tarea de clasificación, las SVMs pertenecen a la categoría de los clasificadores lineales, puesto que inducen separadores lineales o hiperplanos, ya sea en el espacio original de los ejemplos de entrada, si éstos son separables o cuasi-separables (ruido), o en un espacio transformado (espacio de características), si los ejemplos no son separables linealmente en el espacio original

Dado un conjunto de muestras de entrenamiento, cada una marcada como perteneciente a una u otra categoría, el algoritmo de entrenamiento SVM construye un modelo que asigna a las nuevas muestras a una categoría u otra, convirtiéndolo en un clasificador lineal binario. Un modelo SVM es una representación de las muestras como puntos en el espacio, mapeados de modo que las muestras de las dos categorías separadas se dividan por una brecha clara lo más amplia posible. Las nuevas muestras se mapean en ese mismo espacio y se predice (según su ubicación en dicho espacio) a que categoría pertenecen.

Mas formalmente, una máquina de soporte vectorial construye un hiperplano o conjunto de hiperplanos en un espacio de elevadas o infinitas dimensiones. Intuitivamente se consigue una buena separación entre el hiperplano que tiene mayor distancia al punto de datos de entrenamiento más próximo (cuanta más distancia, menor es el error de generalización del clasificador).

Mientras que el problema original puede ser expresando en un espacio dimensional finito, a menudo sucede que los conjuntos a discriminar no son linealmente separables en ese espacio. Por ese motivo, el espacio original de dimensión finita se mapea en un espacio de mayor dimensión, haciendo la separación más sencilla en el nuevo espacio.

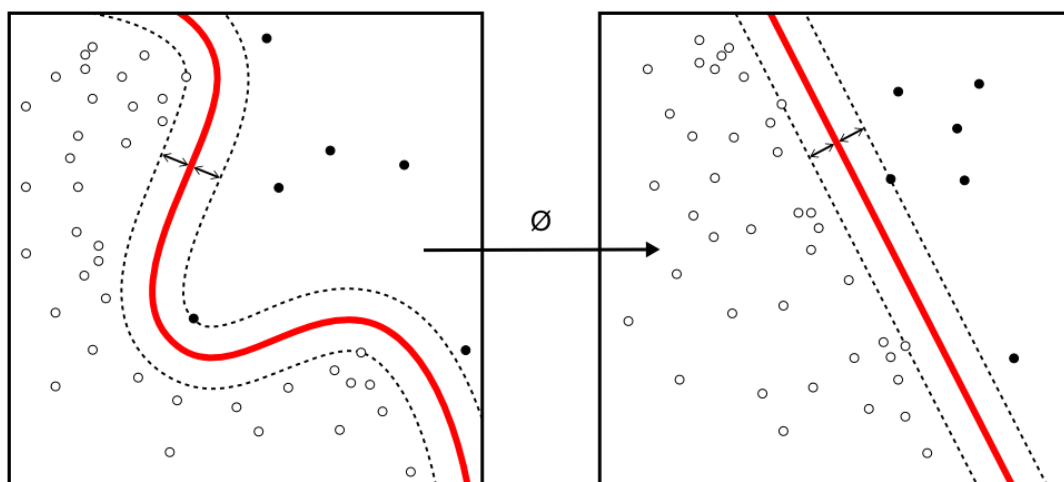


Ilustración 4-3 Transformación a una dimensión superior linealmente separable. El problema de la búsqueda de una función de decisión no lineal en el espacio del conjunto de ejemplos original (espacio de entradas), se puede transformar en un nuevo problema consistente en la búsqueda de una función de decisión lineal (hiperplano) en un nuevo espacio transformado (espacio de características)

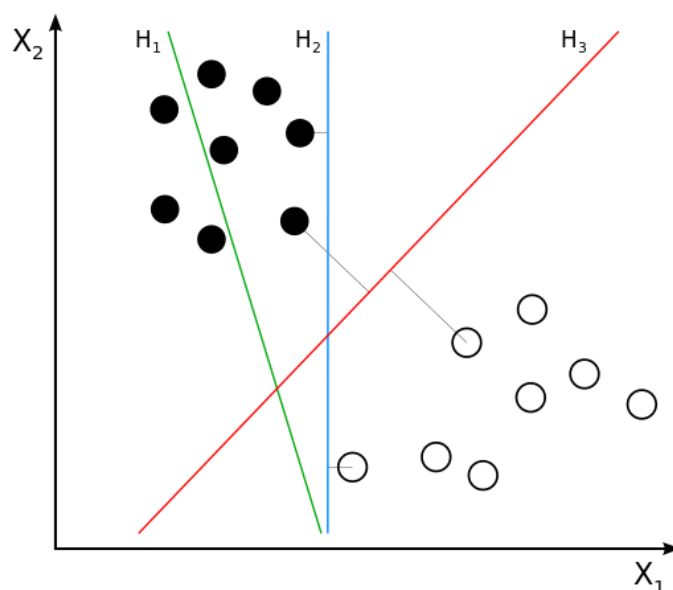


Ilustración 4-4 Hiperplanos a distintos márgenes

H1 => El hiperplano no es capaz de diferenciar las clases.

H2 => El hiperplano es capaz de discernir entre clases, pero con un margen pequeño.

H3 => El hiperplano distingue entre clases con un amplio margen.

Si los datos son linealmente separables, es posible seleccionar dos hiperplanos paralelos que dividan los dos tipos de datos de modo que la distancia entre ellos sea lo más amplia posible. La región delimitada entre los dos hiperplanos se denomina “margen”, y denominamos “hiperplano de margen máximo” al que se encuentra a medio camino entre los dos hiperplanos

paralelos. Estos hiperplanos se definen como

$$\vec{w} \cdot \vec{x} - b = 1$$

y

$$\vec{w} \cdot \vec{x} - b = -1$$

Geoméricamente, la distancia entre estos dos hiperplanos es  $\frac{2}{\|\vec{w}\|}$  para maximizar la distancia entre los planos que se pretende maximizar  $\|\vec{w}\|$ . Para evitar que los puntos de datos caigan dentro del margen, es necesario añadir la siguiente restricción para cada uno de los  $i$  siendo

$$\vec{w} \cdot \vec{x}_i - b \geq 1 \text{ si, } y_i = 1$$

o

$$\vec{w} \cdot \vec{x}_i - b \leq -1 \text{ si, } y_i = -1$$

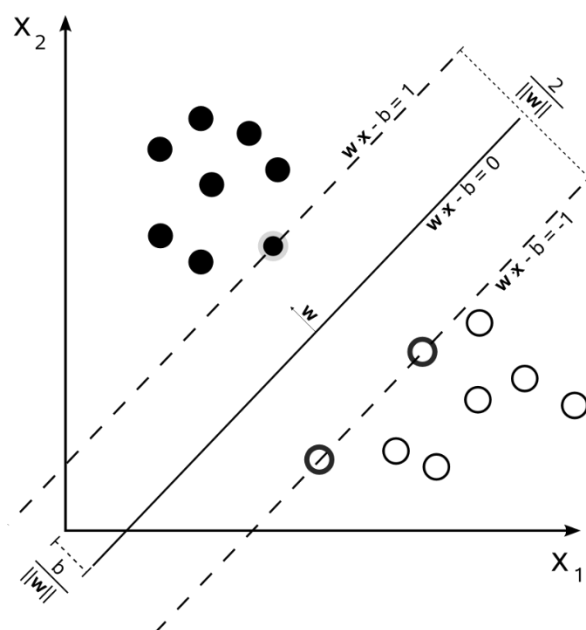


Ilustración 4-5 Representación de hiperplanos paralelos e hiperplano margen.

Estas restricciones establecen que cada punto de datos debe estar en el lado correcto del margen. Una consecuencia fácilmente observable pero importante de esta descripción geométrica, es que el hiperplano de máximo margen está completamente determinado por aquellos  $\vec{x}_i$  que se encuentran más cerca de la construcción. Estos  $\vec{x}_i$  se denominan vectores de soporte

Es de destacar que el trabajo en un espacio de características de mayor dimensión aumenta el error de generalización de las máquinas de vectores de soporte, aunque con suficientes

muestras el algoritmo sigue funcionando correctamente.

Para el cálculo de los vectores de soporte el LSI ha utilizado el software SVMlight. SVMlight es una implementación en C de máquinas de vectores soporte que, en base a un banco de imágenes positivas y negativas adecuadamente preparadas, construye el hiperplano que permite al detector discriminar si la imagen analizada contiene positivos del tipo de entidad buscada.

#### **4.5.2 Configuración del detector basado en HOG**

Como ya hemos adelantado en apartados anteriores, el histograma de gradientes orientados (HOG) [23] es un descriptor de características utilizado en visión por computador y procesamiento de imágenes con el propósito de detectar objetos. La técnica contabiliza la frecuencia en la que se orienta el gradiente en porciones localizadas de una imagen. Para el cálculo de HOG hemos utilizado los métodos implementados en las librerías Opencv

En primer lugar, hemos de definir que es un descriptor. Un Descriptor de características es una representación de una imagen que simplifica la imagen extrayendo la información que se considera útil y descartando el resto. Típicamente, un descriptor convierte una imagen de dimensiones (ancho x alto x nº de canales) en un vector de características de longitud  $n$ . En el caso del descriptor HOG de este ejemplo ilustrativo se utilizará un tamaño de vector =  $64 \times 64 \times 3$ . Es de reseñar que dicho tamaño viene determinado por el detector SVM que vayamos a cargar.

En el descriptor HOG la distribución de las direcciones de los gradientes se utiliza como característica. Los gradientes de una imagen son de gran utilidad en el reconocimiento de objetos ya que la magnitud del gradiente es elevada alrededor de bordes y esquinas. Este tipo de características aportan mucha más información relativa a la forma del objeto que las regiones planas. En las siguientes líneas se explicará brevemente los distintos pasos que conforman el proceso de detección.

##### **Paso 1: Preprocesado:**

Como se ha mencionado en líneas anteriores el descriptor HOG utilizado se calcula en regiones de  $64 \times 64$  dentro de la imagen. Por supuesto la imagen a analizar puede tener cualquier tamaño. La región de  $64 \times 64$  se analizará en las distintas ubicaciones de la imagen, la única restricción es que las regiones a analizar tienen que tener una relación de aspecto fija.

En la Ilustración 4-6 se muestra un ejemplo de preprocesado donde se selecciona una región a analizar, se recorta la imagen y se redimensiona.

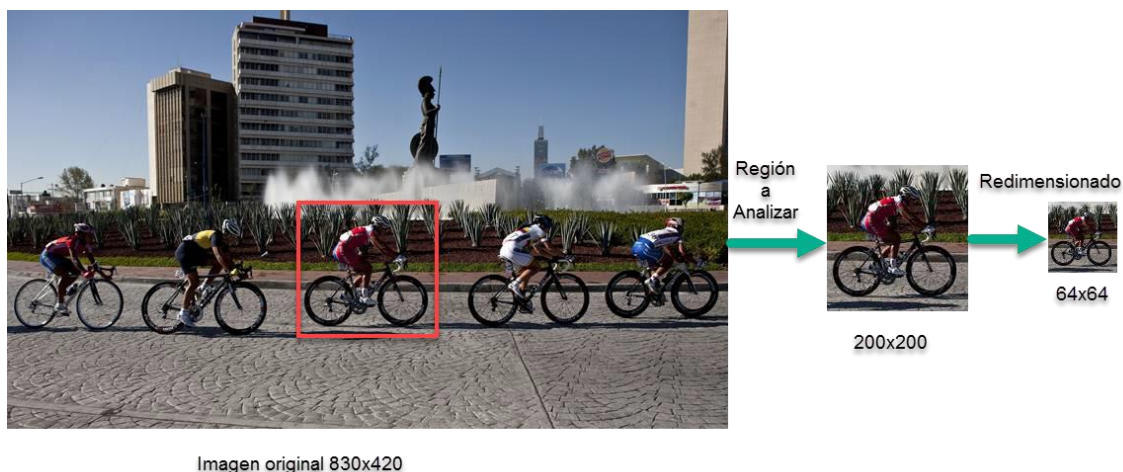


Ilustración 4-6 Ejemplo de preprocesado.

## Paso 2: Calcular las imágenes de degradado.

Tiene como objetivo capturar la información del entorno y silueta de la imagen mediante el cálculo de los gradientes de primer o segundo orden píxel a píxel

Un ejemplo para realizar este cálculo es aplicar kernels horizontales y verticales al histograma.

$$[-1, 0, 1] \text{ y } [-1, 0, 1]^T \text{ Ecuación 1 Kernels horizontales y verticales.}$$



Ilustración 4-7 Ejemplo de cálculo de bordes.

Y por último se calculará la magnitud y dirección del gradiente en cada píxel a según

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

### Paso 3: Cálculo del histograma de gradientes en celdas de 8x8 pixels

En este paso la imagen se divide en celdas de 8x8 y se calcula el histograma de gradientes para cada una de las celdas.

Una de las razones para usar un descriptor de características para describir una región de una imagen es que proporciona una representación compacta de la misma. Una región de  $8 \times 8$  contiene valores para  $8 \times 8 \times 3 = 192$  píxeles. El gradiente para dicha región contiene 2 valores (magnitud y dirección) por cada píxel que suman  $8 \times 8 \times 2 = 128$  números. Estos 128 valores se pueden representar usando un histograma que puede ser almacenado como una matriz de 9 números. No sólo es la representación más compacta sino el cálculo del histograma hace que esta representación sea más robusta al ruido. El histograma es básicamente un vector (o una matriz) de 9 cajas (números) correspondientes a los ángulos 0, 20, 40, 60, 80, 100, 120, 140, 160 y 180 grados. Hay que tomar como consideración que el gradiente se toma sin signo, de esta forma el rango de orientaciones como ya se ha adelantado va desde 0 a  $180^\circ$  dividido en 9 intervalos. El valor de cada uno de estos intervalos se obtendrá de acumular la magnitud de todos estos gradientes asignados al intervalo.

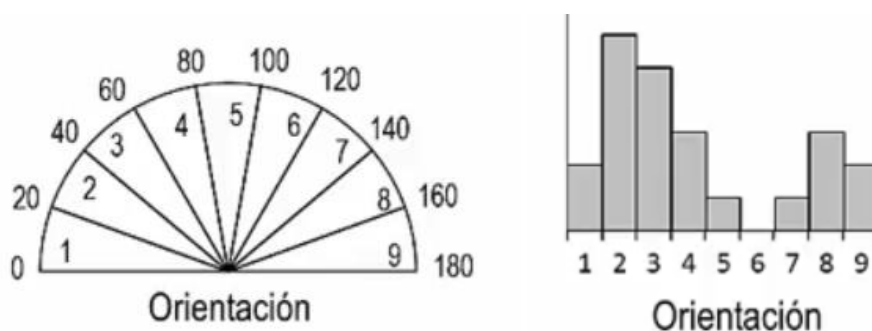


Ilustración 4-8 Diagrama de Orientaciones por intervalos y representación acumulativa por intervalos.

### Paso 4: Normalización.

Los gradientes de una imagen son sensibles a la iluminación. Si se hace que la imagen sea más oscura al dividir todos los valores entre 2, la magnitud del gradiente cambiará a la mitad y, por lo tanto, los valores del histograma se modificarán a la mitad. Idealmente, se pretende que el descriptor no sea dependiente de las variaciones de iluminación. En otras palabras, hay que "normalizar" el histograma para que no se vea afectado por los cambios de iluminación.

Téngase en cuenta que la normalización para HOG no se realiza normalmente para vectores de  $9 \times 1$ . Por ejemplo, se podrían normalizar bloques de 4 histogramas ( $16 \times 16$ ) en vectores de  $36 \times 1$  elementos.

### Paso 5: Cálculo del vector de características HOG.

El último paso consiste en calcular el vector de características final. Para ello no hay más que concatenar todos los vectores ya normalizados calculados en el paso anterior. Este vector hay

que introducirlo a un sistema de reconocimiento basado en aprendizaje supervisado, en el caso de nuestra aplicación serán los clasificadores entrenados para el reconocimiento de ciclistas.

Como ya se adelantó, para el desarrollo de la aplicación se ha hecho uso de las librerías OpenCv, a continuación, se van a describir las clases más importantes utilizadas dentro del proyecto.

- **Clase HOGDescriptor**

Con la finalidad de que las definiciones sean más ilustrativas se va a abordar la explicación con la declaración del objeto que se ha utilizado en el proyecto.

```
hog = new HOGDescriptor(new Size(64, 64), new Size(16, 16) , new  
Size(8, 8), new Size(8, 8), 9, 1, -1.0, HOGDescriptor.L2Hys, 0.2,  
true, HOGDescriptor.DEFAULT_NLEVELS, false);
```

Esta clase implementa el algoritmo desarrollado por Dalal y en su declaración se definen todos los parámetros de configuración. A continuación, se detalla cada uno de estos parámetros en orden de aparición:

- **Tamaño de ventana:** En este campo se define el tamaño de la ventana a analizar. Se mide en píxeles y debe de coincidir con el tamaño de la ventana que se pretende pasar al algoritmo y concordar con el tamaño del bloque y con su desplazamiento. De tal forma que el tamaño de la ventana del HOG debe de coincidir con el tamaño de las imágenes que hayan sido utilizadas en el set de entrenamiento para el SVM.
- **Tamaño del bloque:** Es el tamaño del bloque de normalización y debe de ser proporcional al tamaño de celda. Se mide en píxeles.
- **Desplazamiento del bloque:** Mide en píxeles el desplazamiento del bloque en cada una de las iteraciones del algoritmo. Esta operación se realiza en la fase de normalización y tiene que ser múltiplo del tamaño de la celda.
- **Tamaño de la Celda:** Define en píxeles el tamaño de la celda. A partir de esta agrupación de píxeles se obtienen los vectores gradientes.
- **Número de contenedores:** Define la cantidad de bloques en los que se va a dividir el histograma de gradientes. Este valor ha de ser 9 ya que es el único que admite la librería.
- **Suavizado Gausiano:** Este parámetro lo utiliza el algoritmo antes de comenzar con el análisis y la extracción de los descriptores. En esencia, el efecto mezcla

ligeramente los colores de los píxeles que estén vecinos el uno al otro en un mapa de bits

- **Constante de normalización de bloque.** Es un parámetro que se utiliza en el proceso de normalización. Típicamente se utiliza un valor de 0.2.
- **Corrección gamma.** Parámetro de tipo booleano que es utilizado por el algoritmo en la fase de procesado para disminuir la influencia por iluminación.
- **Número máximo de ventanas de detección.** Es el número máximo de escalados que va a realizar el algoritmo sobre la ventana de detección. Por defecto serán 64.
- **Signo del gradiente.** Parámetro booleano que le indica al algoritmo si se va a utilizar signo para elaborar el gradiente.

- **Método HOGDescriptor::setSVMDetector.**

```
hog.setSVMDetector(descriptors);
```

Este método carga el detector creado en el descriptor. En Android es importante precargar el detector en una matriz de floats del tamaño adecuado determinado por el tamaño de las imágenes pertenecientes al set de entrenamiento. Su único argumento es la matriz de floats.

- **Método HOGDescriptor::detectMultiScale**

Por último, pasaremos a describir el método detectMultiScale. Es el encargado de realizar las detecciones a partir de las imágenes que se le pasan como argumento y de realizar el reescalado de las mismas.

```
hog.detectMultiScale(pMatGris, rects, pesos, 0.5, new Size(8, 8), new  
Size(), 1.05, 2, false);
```

- **Imagen de entrada.** Es la imagen sobre la que se va a realizar la detección. Es posible realizar la detección tanto en imágenes a color como en escala de grises.
- **Localizaciones.** Sobre este parámetro el método devuelve los datos de localización de los objetos detectados sobre la imagen. Guarda la coordenada de la esquina superior izquierda de la ventana en la que se realizó la detección.



- **Pesos detectados.** Este parámetro almacena las puntuaciones de las detecciones que se han obtenido. En la aplicación se utiliza como un parámetro a tener en cuenta en el proceso de filtrado.
- **Umbral de detección.** Parámetro muy importante ya que marca el umbral de distancia entre las características y el hiperplano del SVM. Ajustar bien este parámetro es muy importante para reducir el número de falsos positivos y falsos negativos del detector.
- **Desplazamiento de ventana.** Desplazamiento en píxeles de la ventana de desplazamiento. Dicho desplazamiento se realiza tanto en el eje x como en el eje y hasta recorrer toda la imagen.
- **Relleno.** Parámetro que determina la cantidad de píxeles a añadir en los bordes de la ventana de detección. Este parámetro se configura con la finalidad de mejorar la detección en los bordes de la imagen.
- **Escala.** Factor de escalado que se le realiza a la imagen después de realizar cada análisis de escala.
- **Umbral de Agrupación.** Este parámetro configura el umbral de similitud, es decir, si un mismo objeto ha sido detectado varias veces en múltiples ventanas gracias a este parámetro ciertas detecciones podrán ser filtradas.
- **Usar Agrupamiento de desplazamiento.** Booleano que indica si se debe realizar o no un agrupamiento de desplazamiento.

Ya explicados todos los parámetros para el funcionamiento de la detección cabe incidir en la importancia de algunos de ellos. Como ya hemos adelantado el umbral de detección es un parámetro importantísimo ya que un umbral demasiado bajo puede desencadenar que HOG devuelva un número elevado de falsos positivos mientras que un valor alto de umbral puede conllevar que se pierdan detecciones (falsos negativos).

Otro parámetro importante es el desplazamiento de ventana. La regulación de este parámetro va a depender de la capacidad de computación del dispositivo en el que se va a ejecutar la aplicación. Si se marca un desplazamiento muy pequeño se va a obtener un análisis exhaustivo de la imagen y se va a aumentar la capacidad de detección, pero esta acción va a multiplicar el número de descriptores y a su vez el número de cálculos que se han de realizar.

El último parámetro de importancia es el escalado ya que va a afectar en la detección de la misma manera que el desplazamiento de ventana. Un factor de escala pequeño permitirá analizar una imagen en un número más elevado de escalas, pero aumentando el coste computacional del sistema.

#### 4.5.3 Aplicación para la detección de ciclistas

Una vez detallado el funcionamiento de los algoritmos que permiten realizar la detección se procede a detallar la estructura de la aplicación y el en que consiste cada una de las fases.

## Estructura de la aplicación

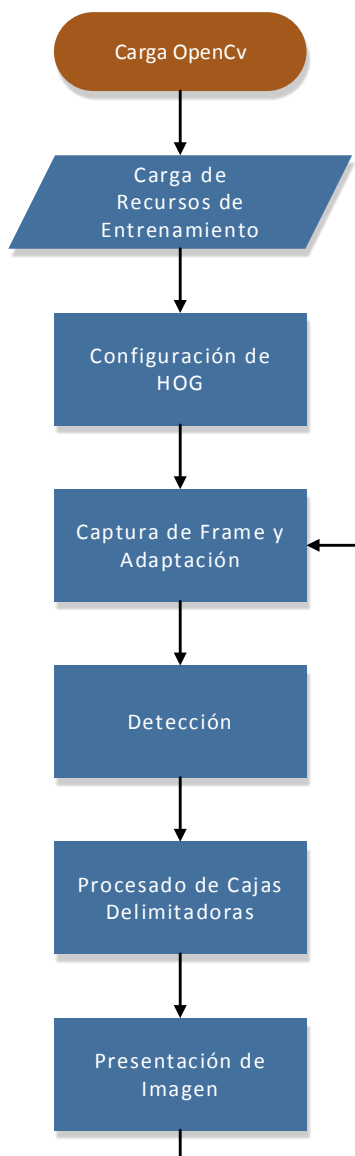


Ilustración 4-9 Diagrama de flujo de la aplicación para la detección de ciclistas.

- **Paso 1: Carga de OpenCv.** Este paso se realiza nada más ejecutar la aplicación, en él se lleva a cabo la carga de la librería OpenCv mediante el uso de la clase

BaseLoaderCallback. Si no hay ninguno problema, una vez finalizada la carga se procederá a activar la vista de cámara de OpenCv.

- **Paso 2: Carga de Recursos de entrenamientos.** Este proceso realiza la carga del fichero procedente de los recursos del sistema y con formato SvmLight, se formatea pasándolo a una cadena y posteriormente se carga en una matriz de floats comprobando que el tamaño es el adecuado.
- **Paso 3: Configuración de HOG.** En esta etapa se configura el objeto HOG con el tamaño y parámetros adecuados para cargar el entrenamiento SVM. Si la matriz de floats tiene el tamaño y contenido correctos el sistema queda preparado para realizar detecciones.
- **Paso 4: Captura de Frame y adaptación.** En la fase de captura de frame se toma la imagen en color desde la cámara del dispositivo, a continuación, se transforma la imagen a escala de grises y finalmente se realiza una reducción de resolución sobre la imagen con el propósito de reducir la carga computacional y el tiempo de procesamiento.
- **Paso 5: Detección.** En la fase de detección se pasa la imagen preformateado al descriptor HOG configurando con el entrenamiento SVM y se configura para que la detección se realiza en modo multi-escala. El algoritmo devuelve dos matrices, una con la posición dentro de la imagen perteneciente a cada una de las detecciones y otra con las puntuaciones generadas por dichas detecciones.
- **Paso 6: Procesado de cajas delimitadoras.** Una vez obtenidas las posiciones de las detecciones se procede a dibujarlas, pero esta operación se realiza sobre la imagen a color capturada y con una resolución más elevada.
- **Paso 7: Presentación de imagen procesada.** Este último paso en el proceso devuelve la imagen a color ya procesada con las detecciones y se muestra en la pantalla del dispositivo. Una vez devuelta se vuelve a ejecutar el paso 4 para seguir con el proceso de detección de manera continua hasta que el usuario decida detener la aplicación.

## 4.6 METODO DE SEGUIMIENTO Y PERSISTENCIA TEMPORAL

Una vez desarrollada la algoritmia para la detección de ciclistas era necesario dotarla de cierta inteligencia. Para ello se desarrolló una segunda aplicación basada en el código ya utilizado por la aplicación base del LSI para el seguimiento de peatones y de coches. El seguimiento o “tracking” trata de dar cierta inteligencia a la detección de objetos y basa su funcionamiento en el análisis de la dinámica del propio objeto para intentar hacer una predicción de donde se encontrará el objeto en los próximos frames y además discernir si es un nuevo objeto el que entra en escena o era uno ya existente. Para dotar de seguimiento y para realizar la asociación de blancos se han empleado el filtro de Kalman y el algoritmo húngaro respectivamente.

### 4.6.1 Filtro de Kalman

El filtro de Kalman [33] se encuentra definido en términos de variables de estado y su solución es calculada de manera recursiva. Sirve para poder identificar el estado oculto (no medible) de un sistema dinámico lineal. Cabe destacar que este sistema funciona además cuando el sistema está sometido a ruido blanco aditivo. En este tipo de filtro, la ganancia  $K$  de realimentación del error es escogida de forma óptima cuando se conocen las varianzas de los ruidos que afectan al sistema. Ya que el Filtro de Kalman es un algoritmo recursivo, este puede correr en tiempo real usando únicamente las mediciones de entrada actuales, el estado calculado previamente y su matriz de incertidumbre.

El filtro se divide en dos fases fundamentales, la de predicción y la de corrección.

La fase de predicción realiza una proyección hacia adelante del estado actual obteniéndose así una predicción del estado. A continuación, en la fase de corrección se añade la medida real a la estimación y se obtiene una estimación mejorada.

Otra característica relevante del filtro de Kalman es que la ganancia crece de forma inversa a el error de medición. Cuando se obtiene un error pequeño, la ganancia de Kalman es más grande y por lo tanto aumenta la ponderación sobre la medida real con respecto a la predicción y por lo tanto confiando menos en ella. En el caso contrario, cuando el error aumenta provoca que la ganancia disminuya y el algoritmo en este caso confíe más en la predicción que en la propia medida.

Por este motivo se ha utilizado el filtro de Kalman para realizar el seguimiento ya que prevé buenos resultados en cualquiera de los casos, aunque hay que tener en cuenta que la distancia en coordenadas reales se realiza a través de una estimación con su correspondiente error que en este caso será acumulativo.

#### 4.6.2 Algoritmo Húngaro

El algoritmo Húngaro [34] es un método de optimización de problemas de asignación. El algoritmo es utilizado para la resolución de problemas de minimización que en nuestro caso se explica a continuación.

Cuando se realiza un seguimiento sobre varios objetos, como ya hemos descrito en el apartado anterior, el filtro de Kalman realiza una serie de predicciones para cada uno de los objetos detectados. En el siguiente frame en el mejor de los casos se vuelven a obtener detecciones para cada uno de los objetos que ya se habían detectado anteriormente y además se cuenta con las predicciones proporcionadas por el filtro de Kalman. Entonces, el problema se reduce a encontrar correctamente que predicción corresponde para cada una de las nuevas detecciones y este problema es el que intenta resolver el algoritmo húngaro. A continuación, se detalla en cinco pasos como el Algoritmo intenta resolver el problema de minimización expuesto y se realizará un ejemplo numérico para ilustrar el caso.

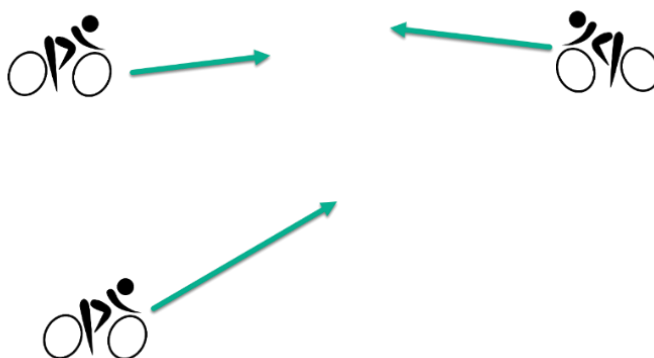


Ilustración 4-10 Ejemplo de detección Frame t. Cada ciclista representa una detección y las flechas ilustran el vector velocidad y sentido.

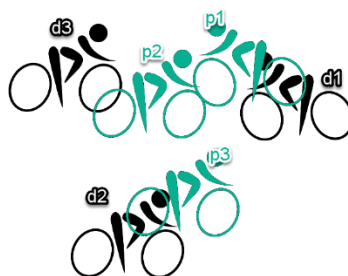


Ilustración 4-11 Ejemplo de detección Frame t+1. Los ciclistas negros representan las detecciones nuevas para el siguiente frame mientras que los ciclistas verdes representan las predicciones proporcionadas por el filtro de Kalman.

- **Paso 1:** En este primer paso se determinará la matriz de costes  $m \times m$  que en este caso viene determinada por  $m$  filas que representan las predicciones en un determinado frame y por otro lado las  $m$  columnas que vienen definidas por las nuevas detecciones encontradas. Una vez construida la matriz se debe de encontrar el elemento más pequeño en cada una de las filas de la matriz. Ejemplo:

$$\begin{array}{c}
 d1 \quad d2 \quad d3 \\
 \begin{array}{l} p1 \\ p2 \\ p3 \end{array} \begin{pmatrix} 10 & 9 & \mathbf{5} \\ 9 & 8 & \mathbf{3} \\ 6 & \mathbf{4} & 7 \end{pmatrix}
 \end{array}$$

- **Paso 2:** Una vez completado el paso anterior se ha de construir una nueva matriz la cual vendrá determinada por los valores resultantes de la diferencia entre cada coste y el valor mínimo de la fila a la cual corresponde cada coste (valor mínimo hallado en el paso anterior). Ejemplo:

$$\begin{array}{c}
 d1 \quad d2 \quad d3 \\
 \begin{array}{l} p1 \\ p2 \\ p3 \end{array} \begin{pmatrix} 5 & 4 & 0 \\ 6 & 5 & 0 \\ 2 & 0 & 3 \end{pmatrix}
 \end{array}$$

- **Paso 3:** El paso tres consiste en construir una nueva matriz siguiendo los dos primeros pasos, pero esta vez aplicándolo a las columnas en vez de a las filas, pero en la matriz resultante del paso 2. Es decir, primero se seleccionan los elementos más pequeños en cada una de las columnas y finalmente se construye la matriz con los valores

resultantes de la diferencia entre cada coste y el valor mínimo de la columna a la cual corresponde cada coste. Esta matriz se denomina "Matriz de costes reducida".

Ejemplo:

$$\begin{array}{c} d1 \quad d2 \quad d3 \\ p1 \begin{pmatrix} 5 & 4 & 0 \end{pmatrix} \\ p2 \begin{pmatrix} 6 & 5 & 0 \end{pmatrix} \\ p3 \begin{pmatrix} 2 & 0 & 3 \end{pmatrix} \end{array} \rightarrow \begin{array}{c} d1 \quad d2 \quad d3 \\ p1 \begin{pmatrix} 3 & 4 & 0 \end{pmatrix} \\ p2 \begin{pmatrix} 4 & 5 & 0 \end{pmatrix} \\ p3 \begin{pmatrix} 0 & 0 & 3 \end{pmatrix} \end{array}$$

- **Paso 4:** Ahora se deben de trazar líneas horizontales o verticales o ambas con el propósito de reducir todos los ceros de la matriz de costes reducidos con el menor número de líneas posibles, si el número de líneas es igual al número de filas o columnas se ha logrado llegar a la solución óptima. Si por el contrario el número de líneas trazadas es inferior al número de filas o columnas se debe de continuar con el siguiente paso. Ejemplo:

$$\begin{array}{c} d1 \quad d2 \quad d3 \\ p1 \begin{pmatrix} 3 & 4 & 0 \end{pmatrix} \\ p2 \begin{pmatrix} 4 & 5 & 0 \end{pmatrix} \\ p3 \begin{pmatrix} 0 & 0 & 3 \end{pmatrix} \end{array}$$

- **Paso 5:** Ahora hay que encontrar el menor elemento de aquellos valores que no se encuentran cubiertos por las líneas del paso 4 y se restará de los elementos que no se encuentran cubiertos por las líneas ya trazadas. A continuación, este mismo valor se sumará a los valores que se encuentren en las intersecciones de las líneas horizontales y verticales. Una vez finalizado este paso se debe volver a ejecutar al paso 4. Ejemplo:

$$\begin{array}{c} d1 \quad d2 \quad d3 \\ p1 \begin{pmatrix} 3 & 4 & 0 \end{pmatrix} \\ p2 \begin{pmatrix} 4 & 5 & 0 \end{pmatrix} \\ p3 \begin{pmatrix} 0 & 0 & 3 \end{pmatrix} \end{array} \quad \begin{array}{c} d1 \quad d2 \quad d3 \\ p1 \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \\ p2 \begin{pmatrix} 1 & 2 & 0 \end{pmatrix} \\ p3 \begin{pmatrix} 0 & 0 & 6 \end{pmatrix} \end{array} \quad \begin{array}{c} d1 \quad d2 \quad d3 \\ p1 \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \\ p2 \begin{pmatrix} 1 & 2 & 0 \end{pmatrix} \\ p3 \begin{pmatrix} 0 & 0 & 6 \end{pmatrix} \end{array}$$

Finalmente, en el ejemplo se observa que se ha conseguido resolver el problema de asignación. La predicción 1 corresponde a la detección 1, la predicción 2 corresponde a la detección 3 y finalmente la predicción 3 tiene correspondencia con la detección 2.

#### 4.6.3 Sistema de estimación de Distancias.

El sistema de estimación de distancias que se utiliza en la aplicación para el seguimiento de ciclistas es el mismo que había sido desarrollado previamente en el laboratorio de sistema inteligentes ya se utilizaba en el resto de la aplicación, desarrollado por Victor Cancho Arnesto como Trabajo Fin de Grado [35] y las mejoras incorporadas por Jesús Urdiales Parra [36] también en su Trabajo Fin de Grado y basado en el método descrito por Arturo de la Escalera [37]. Este método realiza la estimación de distancia entre el dispositivo y el ciclista utilizando un punto de contacto entre objeto y suelo. Este punto de contacto viene determinado por el centro inferior de la caja delimitadora tal y como se muestra en la Ilustración 4-12.



Ilustración 4-12 Punto de contacto entre suelo y objeto.

El cálculo del dispositivo con respecto al suelo se basa en el cálculo del vector de gravedad que a su vez es calculado mediante los datos proporcionados por el acelerómetro del dispositivo. Este dato ha de tener en cuenta ya que cuando el dispositivo se encuentre en una zona de pendiente, el vector de gravedad cambiará ocasionando un error en la estimación de distancia. Conociendo la dirección a la que apunta la gravedad descompuesta en cada eje, el algoritmo calcula los distintos ángulos de Euler necesarios para realizar las correcciones en el cálculo de las distancias. Estas correcciones son muy importantes ya que una pequeña inclinación de la cámara se traduce en una gran diferencia en la imagen por lo que el algoritmo actúa en consecuencia.



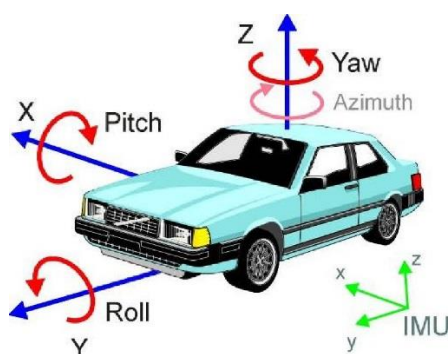


Ilustración 4-13 Ángulos de Euler Pitch y Roll.

La estimación de distancias se realiza calculando la diferencia en píxeles entre el punto de contacto y el centro de la imagen, ya que esta no presenta error alguno como consecuencia de la curvatura de la lente. Para este cálculo como ya se ha explicado anteriormente también se tiene en cuenta la elevación del dispositivo con respecto al suelo, parámetro configurable dentro de la aplicación y que se ha de introducir correctamente para disminuir al máximo el error.

#### 4.6.4 Aplicación para el seguimiento de ciclistas

Una vez detallado el funcionamiento de los algoritmos que permiten realizar el seguimiento de ciclistas se procede a detallar la estructura de la aplicación y el en que consiste cada una de las fases.

Es de destacar que para la realización de esta parte de la aplicación se ha empleado un código desarrollado previamente en el laboratorio de... del alumno Jesús Urdiales Parra [36] realizado como Trabajo Fin de Grado y cuya aplicación consistía en realizar seguimiento de peatones y de vehículos. La adaptación ha consistido en incluir todos los métodos necesarios para realizar una carga fiable de entrenamientos SVM, llevar a cabo la detección mediante HOG y SVM y adaptar el código para su correcto funcionamiento.

## Estructura de la aplicación

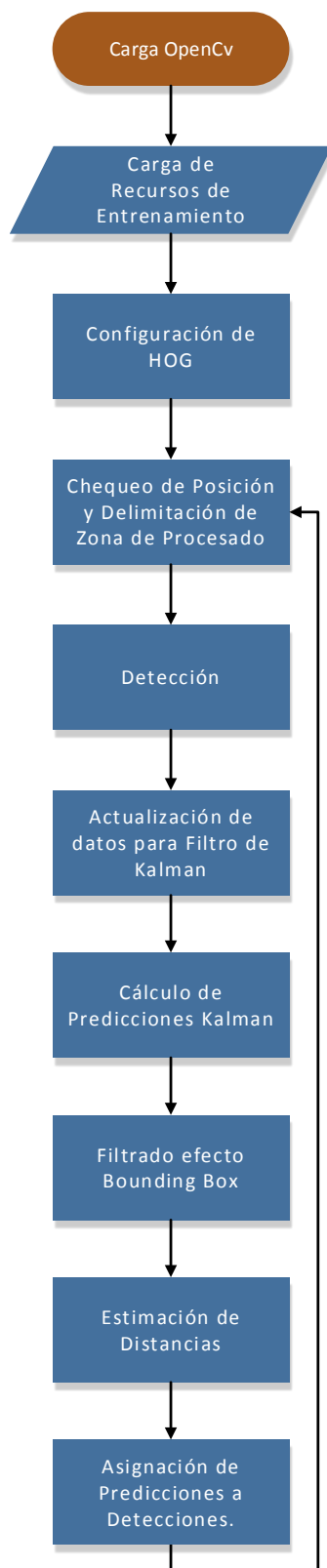


Ilustración 4-14 Diagrama de flujo de la aplicación para el seguimiento de ciclistas.

- **Paso 1: Carga de OpenCv.** Este paso se realiza nada más ejecutar la aplicación, en él se lleva a cabo la carga de la librería OpenCv mediante el uso de la clase BaseLoaderCallback. Si no hay ningún problema, una vez finalizada la carga se procederá a activar la vista de cámara de OpenCv.
- **Paso 2: Carga de Recursos de entrenamientos.** Este proceso realiza la carga del fichero procedente de los recursos del sistema y con formato SvmLight, se formatea pasándolo a una cadena y por último se carga en una matriz de floats comprobando que el tamaño es el adecuado.
- **Paso 3: Configuración de HOG.** En esta etapa se configura el objeto HOG con el tamaño y parámetros adecuados para cargar el entrenamiento SVM. Si la matriz de floats tiene el tamaño y contenido correctos el sistema queda preparado para realizar detecciones.
- **Paso 3: Chequeo de posición y delimitación de la zona de procesado.** En esta etapa se hace uso de la biblioteca de estimación de distancias del dispositivo para calcular los ángulos de Pitch y Roll. Este proceso se realiza para comprobar que dichos ángulos se encuentran dentro de los márgenes de operación. Si el dispositivo se encuentra demasiado inclinado se mostrará una caja de procesado en color rojo que indica la imposibilidad de realizar la detección y el seguimiento sobre los ciclistas. En caso de encontrarse dentro de los límites se mostrará en verde un recuadro que indica los límites del área de procesado configurado.
- **Paso 5: Detección.** En la fase de detección se le pasa la imagen preformateada al descriptor HOG configurando con el entrenamiento SVM y se configura para que la detección se realiza en modo multi-escala. El algoritmo devuelve dos matrices, una con la posición dentro de la imagen perteneciente a cada una de las detecciones y otra con las puntuaciones generadas por dichas detecciones.
- **Paso 6: Actualización de datos para el filtro de Kalman.** Como ya se explicó en el apartado 4.6.1 las matrices de covarianza y de estados son dependientes del tiempo por lo que hay que alimentarlas con el tiempo transcurrido entre frames para su actualización. Una vez actualizadas dichas matrices el filtro se encuentra preparado para realizar las predicciones oportunas.
- **Paso 7: Cálculo de predicciones mediante filtro de Kalman.** Una vez actualizadas las matrices de covarianza y estados, el filtro de Kalman estima la posición de los

objetos en función de su posición, rumbo y velocidad. Nótese que se ha utilizado la biblioteca jKalman [38] de código abierto para la realización de los cálculos de predicción y se ha estimado como constante la velocidad de los tracks.

- **Paso 8: Filtrado efecto Bounding Box.** El efecto Bounding Box o de superposición de la caja de detección o delimitadora consiste en que para un único positivo el sistema refleja varios resultados positivos superponiendo unas cajas con otras. Para ello el algoritmo basa su funcionamiento en buscar las detecciones solapadas, eliminando aquella con menor peso (la cual es calculada por el propio algoritmo para cada detección).
- **Paso 9: Estimación de Distancias.** En este paso se hace uso del algoritmo de estimación de distancias (apartado 4.6.3) para cada una de las detecciones y se guarda en una lista para su posterior uso.
- **Paso 10: Asignación de predicciones a detecciones.** Finalmente se crea la matriz de costes con los datos proporcionados por las nuevas detecciones ya filtradas y las predicciones provenientes del filtro de Kalman. Una vez aplicado el algoritmo Húngaro (apartado 4.6.2) se obtienen las asociaciones pertinentes entre detecciones y predicciones. Como es obvio el número de detecciones no ha de ser igual al de predicciones, dentro de esta etapa se deciden que tracks son eliminados o si por el contrario es necesario crear nuevos tracks.

## 5. EVALUACIÓN: PRUEBAS, RESULTADOS Y LIMITACIONES

A continuación, se van a discutir las pruebas efectuadas al bloque de detección de ciclistas y al bloque de seguimiento. Para ambos bloques se comentarán tanto resultados como el propio funcionamiento de los mismos. También se dará cuenta de las limitaciones del sistema.

### 5.1 LIMITACIONES.

Como ya hemos introducido anteriormente la máquina de soporte vectorial, a pesar de ser un sistema puntero a la hora de crear clasificadores para la detección, cuenta con una serie de limitaciones intrínsecas al propio sistema. Una de las principales limitaciones es el elevado coste computacional que requiere la detección por este método. Debido a ello se han adoptado las medidas ya explicadas anteriormente para la reducción de dicho coste como la conversión de los frames a escala de gris o la disminución de la resolución de las imágenes.

Otra limitación importante radica en su procedimiento de optimización ya que es difícil de implementar y limita su uso para problemas con muchas muestras como es el caso. Para acotar y evaluar la problemática se han utilizado tres clasificadores distintos para la detección de los ciclistas.

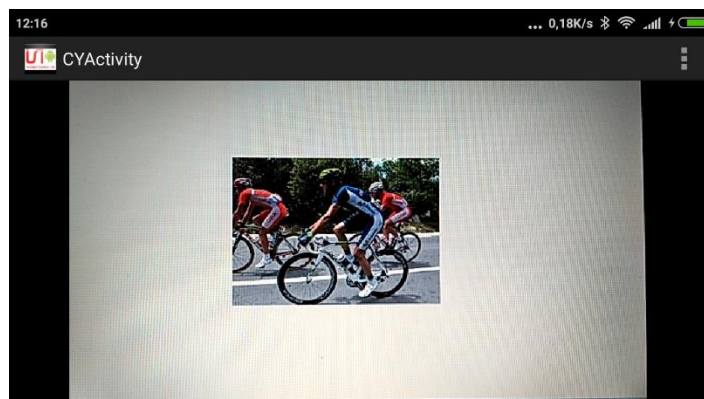


Ilustración -1 Limitaciones -Problema de optimización

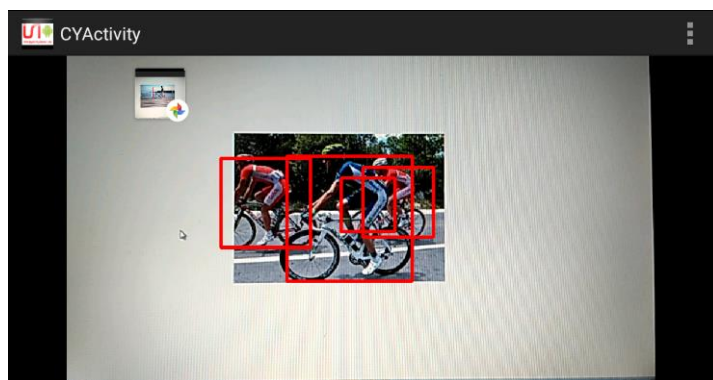


Ilustración 5-2 Limitaciones -Problema de optimización

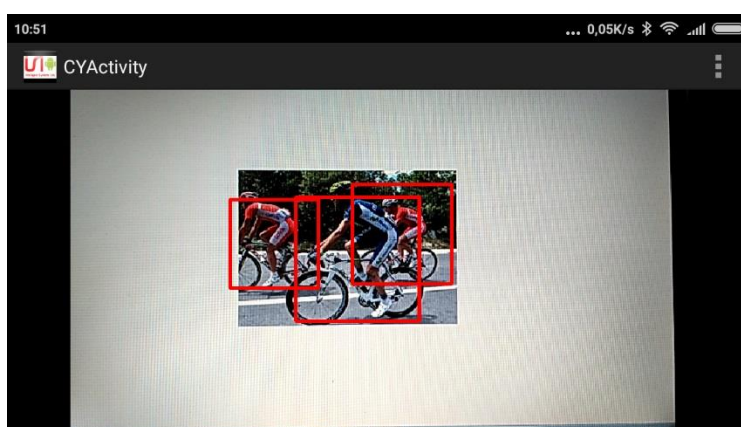


Ilustración 5-3 Limitaciones – Código optimizado correctamente

Aunque el método de detección HOG es muy robusto frente a cambios de iluminación y fondos desfavorables, es digno de reseñar que se han encontrado dificultades para la detección de ciclistas en ciertos entornos como se muestra en las siguientes imágenes.

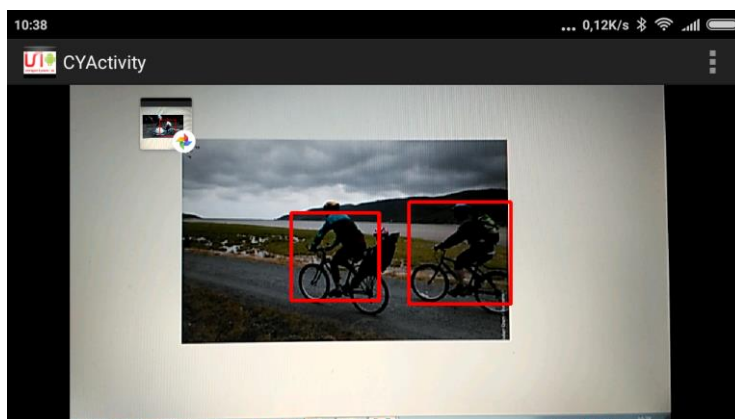


Ilustración 5-4 Limitaciones – Detección con Fondo e iluminación adversa.

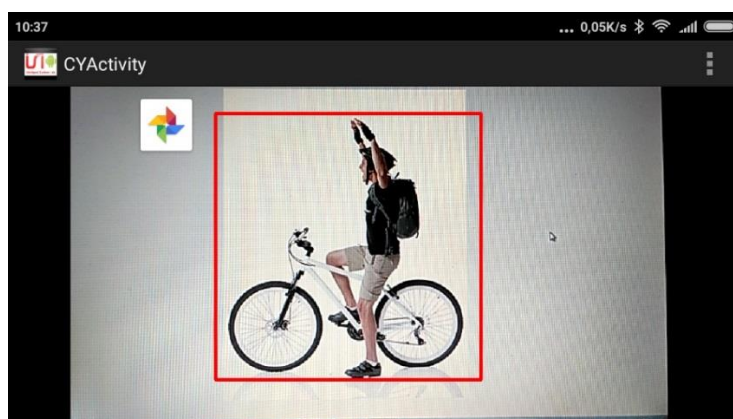


Ilustración 5-5 Limitaciones – Detección con Fondo e iluminación óptima.

Por último, uno de los principales problemas típicos encontrados durante el desarrollo es el problema de superposición de la caja de detección o delimitadora (overlapping bounding box problem). El efecto de este error es que para un único positivo el sistema refleja varios resultados positivos superponiendo unas cajas con otras. Hay varias maneras de solventar este problema. Triggs et al [23]. Sugiere utilizar el algoritmo Mean-Shift para detectar múltiples modos en el espacio del cuadro delimitador utilizando las coordenadas (x, y) así como el logaritmo de la escala actual de la imagen, pero este método tiene un coste computacional muy elevado. En el apartado de detección se ha observado que una buena definición de todos los parámetros de HOG para la detección con el entrenador SVM reduce cuantiosamente la reproducción de este efecto no deseado, no obstante, aunque ya se ejecuta un algoritmo basado en buscar las detecciones solapadas, eliminando las de menor peso, el efecto se reproduce, aunque en muy pocas ocasiones. A continuación, se muestra un ejemplo claro de overlapping reproducido en el sistema.

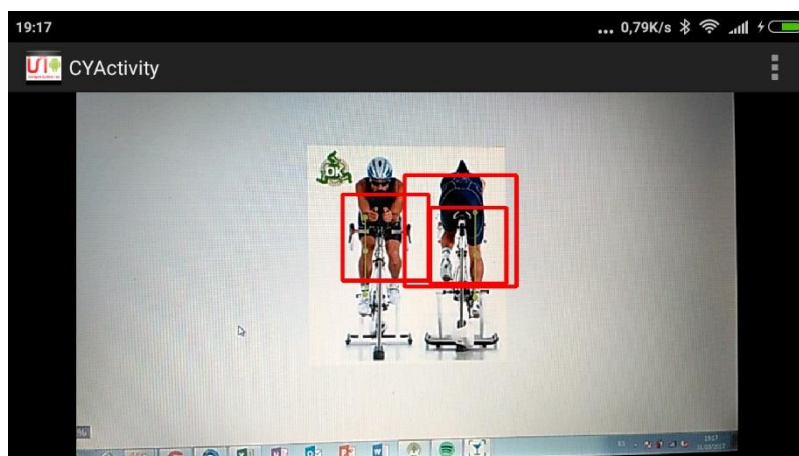


Ilustración 5-6 Limitaciones – Problema de Overlapping.

## 5.2 METODOLOGÍA PARA LA EVALUACIÓN

Con el fin de evaluar los distintos clasificadores y sus valores de umbral se ha optado por utilizar como método de evaluación la curva ROC (Receiver Operating Characteristic) [39].

La curva ROC tiene como objetivo seleccionar el clasificador que mejor se comporte en unas circunstancias determinadas. Un modelo de clasificación como el nuestro es una función que permite decidir que instancias de un conjunto tienen relación entre sí o no por pertenecer a un mismo tipo. El resultado del clasificador es un valor binario determinado por un valor de umbral (en nuestro caso la posición del resultado del clasificador SVM con respecto al hiperplano creado).

Los resultados obtenidos se clasificarán como positivos (p) o negativos (n). Se pueden dar cuatro posibles resultados a partir del clasificador binario. Si el resultado de la búsqueda es p y el valor dado también es p, entonces se categoriza como un verdadero positivo (VP). Por otro lado, obtendremos un verdadero negativo cuando el resultado de la exploración sea n y el valor dado también sea n. Otro escenario que contemplar es el de los falsos positivos (FP) en el cual obtendremos un resultado en la exploración de p cuando se ha introducido un valor real n. Como último escenario posible se puede obtener también un falso negativo (FN) en el cual obtendremos un resultado n ante un valor real p.

Definiendo el experimento a partir de P instancias positivas y N instancias negativas se obtiene una matriz de confusión como la que se muestra en la siguiente figura:

		Valor en la realidad		
		<i>p</i>	<i>n</i>	total
Predicción outcome	<i>p'</i>	Verdaderos Positivos	Falsos Positivos	P'
	<i>n'</i>	Falsos Negativos	Verdaderos Negativos	N'
total		P	N	

Ilustración 5-7 Matriz de confusión.



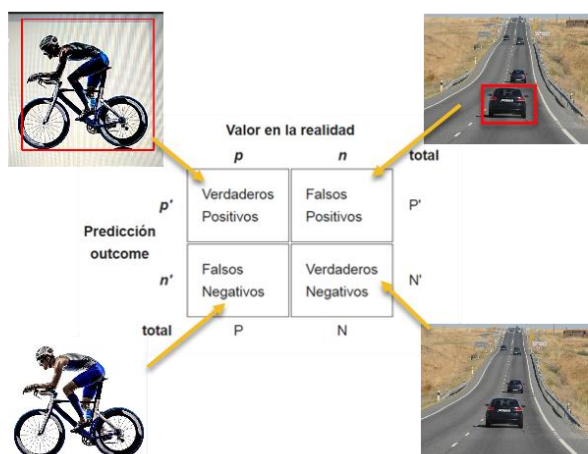


Ilustración 5-8 Ejemplo ilustrativo de la matriz de confusión.

La curva es una representación gráfica de la sensibilidad o razón de verdaderos positivos (VPR) frente a la especificidad o razón de falsos positivos. Para poder representar la curva ROC se ha de tener en cuenta la siguiente terminología.

- Verdaderos Positivos (VP) o también éxitos.
- Verdaderos Negativos (VN) o también rechazos correctos.
- Falsos Positivos (FP) o también falsas alarmas o Error de tipo I.
- Falsos Negativos (FN) o Error de tipo II.
- Sensibilidad o Razón de Verdaderos Positivos (VPR).

$$VPR = \frac{VP}{P} = \frac{VP}{(VP + FN)}$$

- Ratio o Razón de Falsos Positivos (FPR).

$$FPR = \frac{FP}{N} = \frac{FP}{(FP + VN)}$$

- Exactitud (ACC).

$$ACC = \frac{(VP + VN)}{(P + N)}$$

- Especificidad (SPC) o Razón de verdaderos negativos.

$$SPC = \frac{VN}{N} = \frac{VN}{(FP + VN)} = 1 - FPR$$

El espacio ROC se define por FPR y VPR como ejes x e y respectivamente, y representa los intercambios entre verdaderos positivos (beneficios) y falsos positivos (costes). Dado que VPR es equivalente a sensibilidad y FPR es igual a 1-especificidad, el gráfico ROC también es conocido como la representación de sensibilidad frente a (1-especificidad). Cada resultado de predicción o instancia de la matriz de confusión representa un punto en el espacio ROC.

El mejor método posible de predicción se situaría en un punto en la esquina superior izquierda, o coordenada (0,1) del espacio ROC, representando un 100% de sensibilidad (ningún falso negativo) y un 100% también de especificidad (ningún falso positivo). Por el contrario, una clasificación totalmente aleatoria (o predicción aleatoria) daría un punto a lo largo de la línea diagonal, denominada línea de no-discriminación, desde el extremo inferior izquierdo hasta la esquina superior derecha.

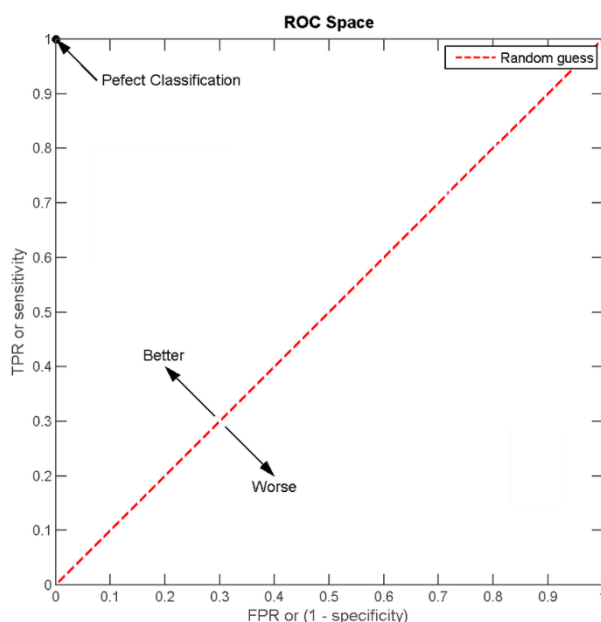


Ilustración 5-9 Espacios curva ROC.

La diagonal divide el espacio ROC. Los puntos por encima de la diagonal representan los buenos resultados de clasificación (mejor que el azar), los resultados encima de la diagonal corresponden a unos resultados completamente aleatorios y por último los puntos por debajo de la diagonal corresponderían a un mal clasificador (peor que el azar). La curva se dibuja mediante los resultados de un clasificador determinado, al ir variando el umbral de decisión, permitiendo además de comparar resultados entre clasificadores, identificar el punto óptimo de umbral (aquel más cercano al punto ideal (0,1)).

El indicador más utilizado para la evaluación del clasificador es el área bajo la curva o AUC. Este índice se puede interpretar como la probabilidad de que un clasificador ordenará o

puntuará una instancia positiva elegida aleatoriamente más alta que una negativa.

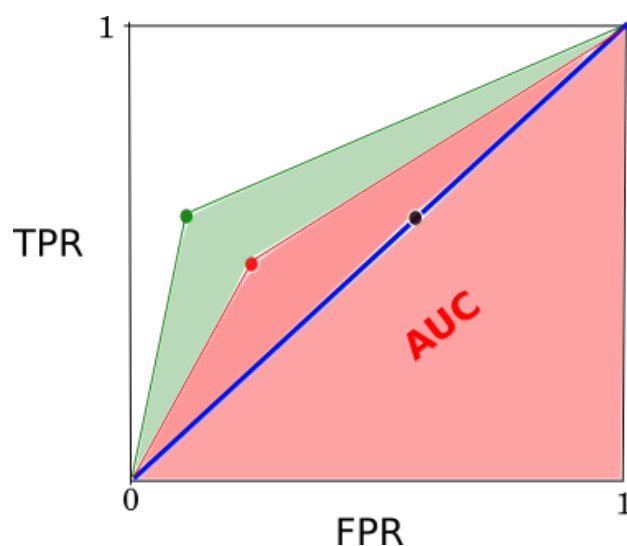


Ilustración 5-10 Construcción de la curva ROC

La razón de verdaderos positivos (TPR) y la razón de falsos positivos (FPR) se proyectan en un plano y luego se calcula el área bajo la curva (AUC) teniendo en cuenta los valores TPR y FPR del clasificador a evaluar, y los puntos extremos (0,0) (1,1).

La ilustración muestra el ejemplo de dos clasificadores en colores verde y rojo y su AUC. Como se puede ver el clasificador verde tiene un AUC mayor.

A modo de guía para interpretar las curvas ROC se han establecido los siguientes intervalos para los valores de AUC:

Valor del Test	Especificidad
(0.5, 0.6)	Test malo
(0.6, 0.75)	Test regular
(0.75, 0.9)	Test bueno
(0.9, 0.97)	Test muy bueno
(0.97, 1)	Test excelente

Tabla 5-1 Interpretación valores AUC.

### 5.3 BANCO DE PRUEBAS PARA LA DETECCIÓN.

Como ya se anticipaba en el apartado anterior, debido a la complejidad de las distintas formas que pueden adoptar los ciclistas dependiendo de la posición en la que se dispongan, se optó por realizar tres entrenamientos distintos dependiendo de la posición de los ciclistas. Se seleccionaron tres bancos de imágenes para realizar cada evaluación sobre cada uno de los clasificadores y un banco de imágenes de verdaderos negativos común a los tres clasificadores. A su vez se han ejecutado las pruebas para 4 umbrales distintos para cada clasificador.

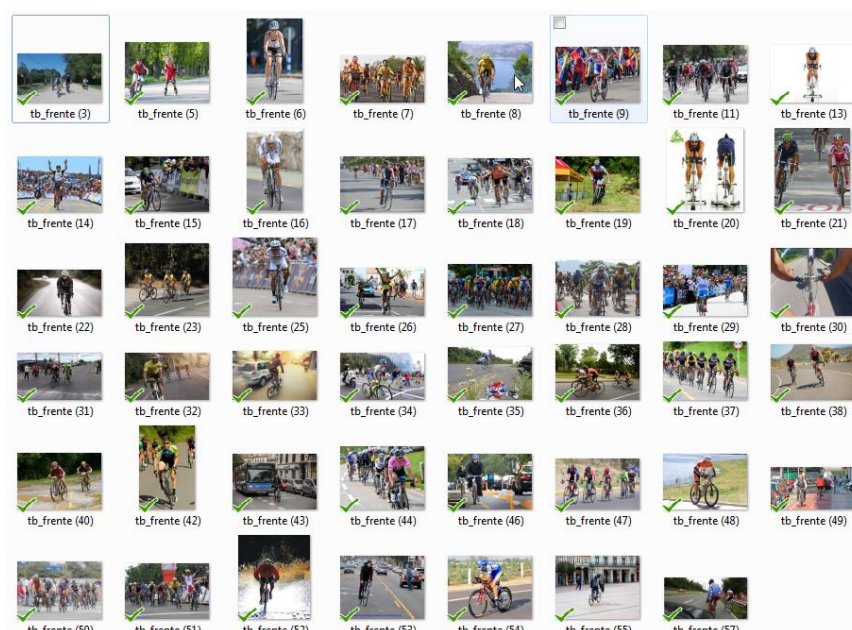


Ilustración 5-11 Imágenes Banco de Pruebas (Frente)

El banco de pruebas utilizado consistió en un pc, en el cual se mostraron las imágenes estáticas. El dispositivo móvil se colocó a una distancia y orientación constantes mediante el uso de un trípode para que la única variable en el experimento fuera el umbral de detección. El dispositivo móvil detectó las imágenes y las capturó para dejar evidencia de la prueba y por último mediante conteo manual se obtuvieron los resultados del experimento.



Ilustración 5-12 Banco de Pruebas.

1. Imágenes Seleccionadas para el banco de pruebas.
2. Condiciones de distancia y orientación constantes.
3. Detección mediante aplicación desarrollada para Android y ejecutada en Smartphone
4. Toma de resultados de las pruebas

### 5.3.1 Resultados para el entrenamiento para ciclistas vistos desde atrás o de frente.

Para realizar las pruebas del clasificador se ha utilizado una batería de 47 imágenes con 87 eventos positivos y se han evaluado para cuatro valores distintos de umbral obteniéndose los resultados que se muestran en la siguiente tabla.

Umbral Central ( $U_c$ )= 0, 42	Positivos	Falsos Positivos	Falsos Negativos	Verdaderos Negativos
	80	12	7	33
Umbral = $U_c + 0,02$	Positivos	Falsos Positivos	Falsos Negativos	Verdaderos Negativos
	48	6	39	33
Umbral Central = $U_c + 0,06$	Positivos	Falsos Positivos	Falsos Negativos	Verdaderos Negativos
	21	2	66	33
Umbral Central = $U_c - 0,02$	Positivos	Falsos Positivos	Falsos Negativos	Verdaderos Negativos
	84	48	3	30

Tabla 5-2 Resultados Vista Frontal y Trasera.

### 5.3.2 Resultados para el entrenamiento para ciclistas vistos de perfil.

Para realizar las pruebas del clasificador se ha utilizado una batería de 48 imágenes con 63 eventos positivos y se han evaluado para cuatro valores distintos de umbral obteniéndose los resultados que se muestran en la siguiente tabla.

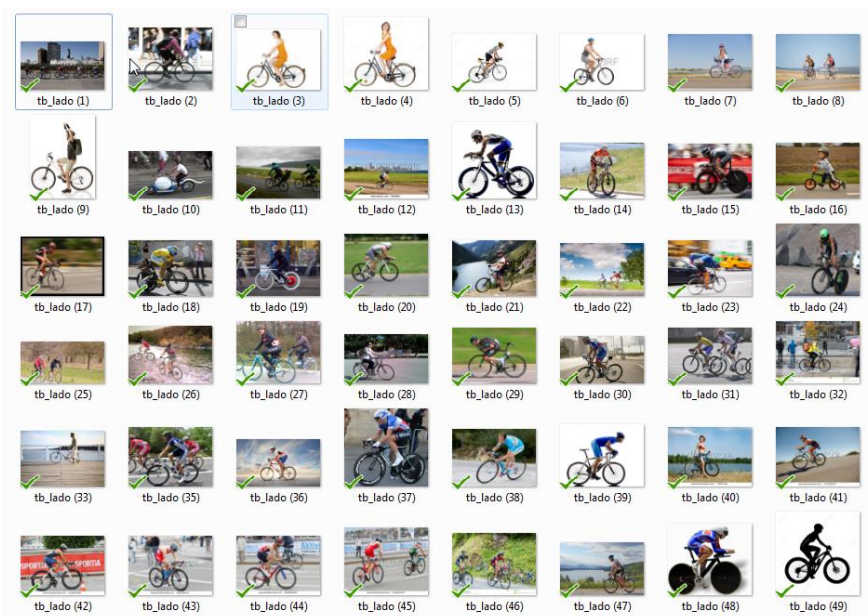


Ilustración 5-13 Imágenes Banco de Pruebas (Perfil)

Umbral Central ( $U_c$ )= 0, 5	Positivos	Falsos Positivos	Falsos Negativos	Verdaderos Negativos
	60	2	3	33
Umbral = $U_c + 0,1$	Positivos	Falsos Positivos	Falsos Negativos	Verdaderos Negativos
	47	2	16	33
Umbral Central = $U_c + 0,3$	Positivos	Falsos Positivos	Falsos Negativos	Verdaderos Negativos
	14	0	49	33
Umbral Central = $U_c - 0,1$	Positivos	Falsos Positivos	Falsos Negativos	Verdaderos Negativos
	62	28	1	26

Tabla 5-3 Resultados vista de Perfil.



### 5.3.3 Resultados para el entrenamiento para ciclistas vistos de perfil y frente.

Para realizar las pruebas del clasificador se ha utilizado una batería de 40 imágenes con 64 eventos positivos y se han evaluado para cuatro valores distintos de umbral obteniéndose los resultados que se muestran en la siguiente tabla.

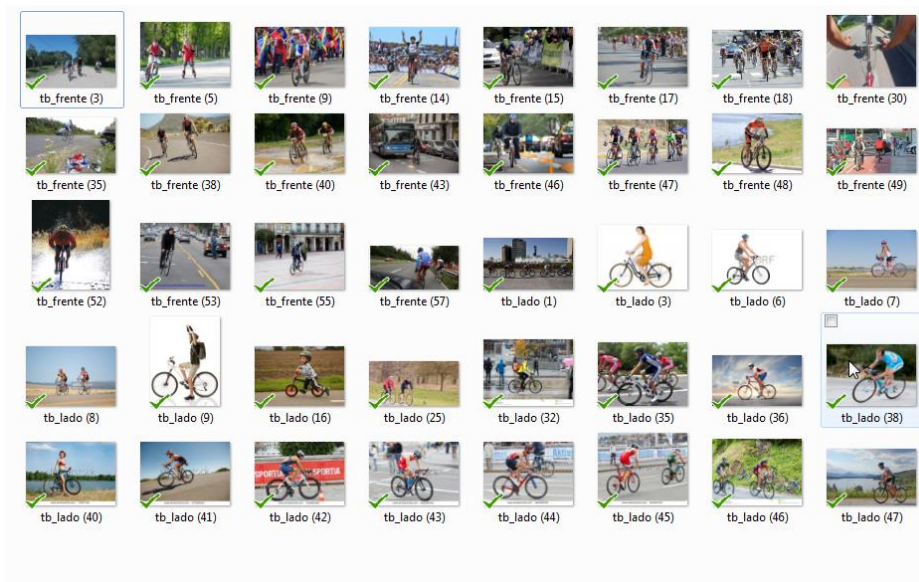


Ilustración 5-14 Imágenes Banco Pruebas (Perfil y Frente)

Umbral Central ( $U_c$ )= 0, 75	Positivos	Falsos Positivos	Falsos Negativos	Verdaderos Negativos
	32	7	32	33
Umbral = $U_c + 0,05$	Positivos	Falsos Positivos	Falsos Negativos	Verdaderos Negativos
	22	4	42	33
Umbral Central = $U_c + 0,15$	Positivos	Falsos Positivos	Falsos Negativos	Verdaderos Negativos
	14	1	50	33
Umbral Central = $U_c - 0,05$	Positivos	Falsos Positivos	Falsos Negativos	Verdaderos Negativos
	39	16	25	31

Tabla 5-4 Resultados mixto (Frente y Perfil).

#### 5.3.4 Conclusiones para la detección.

Con el fin de evaluar los distintos clasificadores (método de evaluación descrito en el apartado 5.2) con sus respectivos niveles de umbral se han calculado para cada uno de ellos la sensibilidad y la especificidad obteniéndose los siguientes resultados.

Valor de Umbral	Sensibilidad	1- Especificidad
VP+ (0,42)	0,91954023	0,266666667
VP+ (0,44)	0,551724138	0,153846154
VP+ (0,48)	0,24137931	0,153846154
VP+ (0,40)	0,965517241	0,615384615

Tabla 5-5 Sensibilidad y Especificidad para clasificador de frente.

Valor de Umbral	Sensibilidad	1-Especificidad
VP+ (0,5)	0,952380952	0,057142857
VP+ (0,6)	0,746031746	0,057142857
VP+ (0,8)	0,222222222	0,057142857
VP+ (0,40)	0,984126984	0,518518519

Tabla 5-6 Sensibilidad y Especificidad para clasificador de perfil.

Valor de Umbral	Sensibilidad	1-Especificidad
VP+ (0,75)	0,5	0,175
VP+ (0,8)	0,34375	0,108108108
VP+ (0,9)	0,21875	0,108108108
VP+ (0,7)	0,609375	0,651685393

Tabla 5-7 Sensibilidad y Especificidad para clasificador mixto (perfil y frente).

Una vez obtenidos los resultados de sensibilidad y especificidad se calcularon las curvas ROC para cada uno de los clasificadores



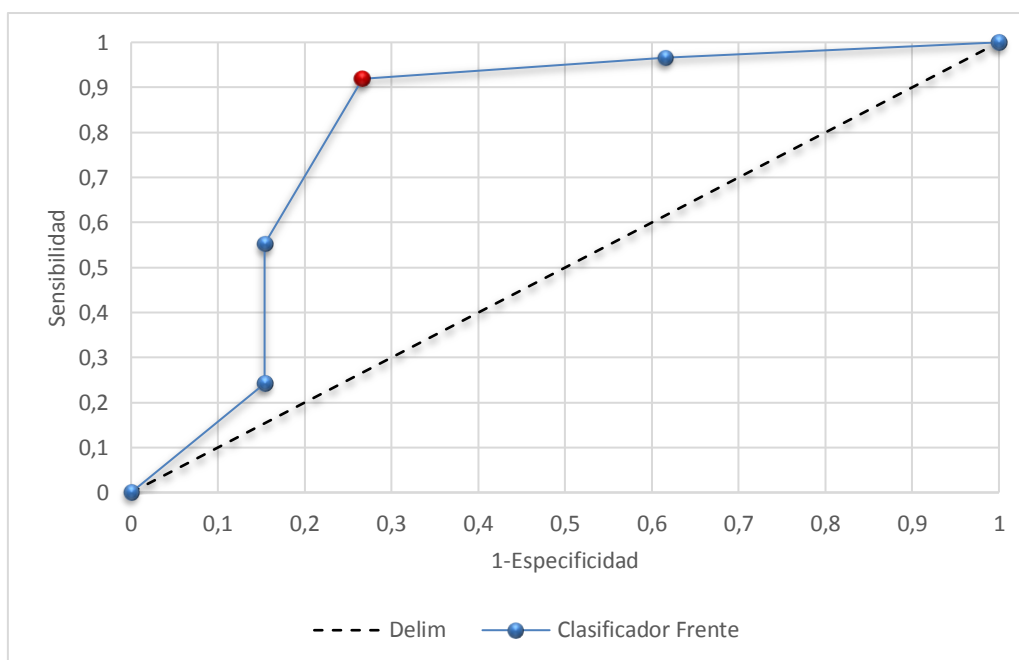


Ilustración 5-15 Banco de Pruebas: Evaluación de los diferentes niveles de umbral para el clasificador de detección de ciclistas de frente en función de la sensibilidad y la especificidad. Para la evaluación de cada clasificador se empleó el banco de imágenes de prueba correspondiente y 4 valores de umbral distintos. La línea delimitadora diagonal representa el límite de aleatoriedad para la evaluación del clasificador. El punto rojo indica el mejor resultado obtenido.

Como se observa en la Ilustración 5-15, para el clasificador para la detección de ciclistas de frente, a simple vista el mejor resultado se obtiene para un valor de umbral de 0.42 (marcado en rojo) ya que como se explicó en el apartado 5.2 el mejor método posible de predicción se situará lo más próximo a la coordenada (0,1).

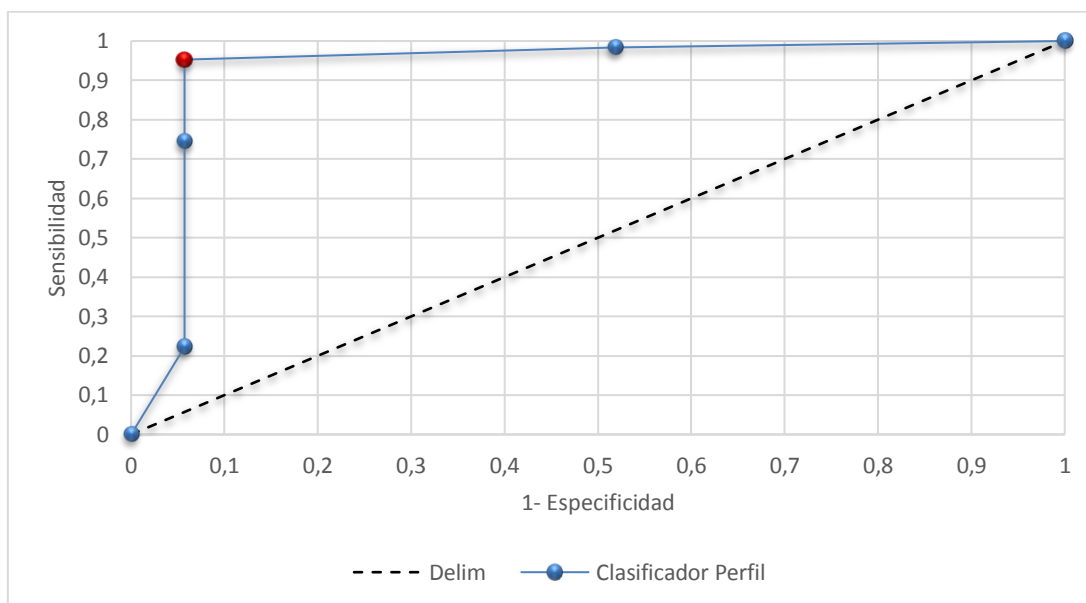


Ilustración 5-16 Banco de Pruebas: **Evaluación de los diferentes niveles de umbral para el clasificador de detección de ciclistas de perfil en función de la sensibilidad y la especificidad.** Para la evaluación de cada clasificador se empleó el banco de imágenes de prueba correspondiente y 4 valores de umbral distintos. La línea delimitadora diagonal representa el límite de aleatoriedad para la evaluación del clasificador. El punto rojo indica el mejor resultado obtenido.

En la Ilustración 5-16 se observa que, para el clasificador para la detección de ciclistas de perfil, el mejor resultado se obtiene para un valor de umbral de 0.5 (marcado en rojo). Los resultados para este valor de umbral son excelentes ya que la tasa de verdaderos positivos es muy alta mientras que la de falsos positivos se mantiene en unos valores significativamente bajos.

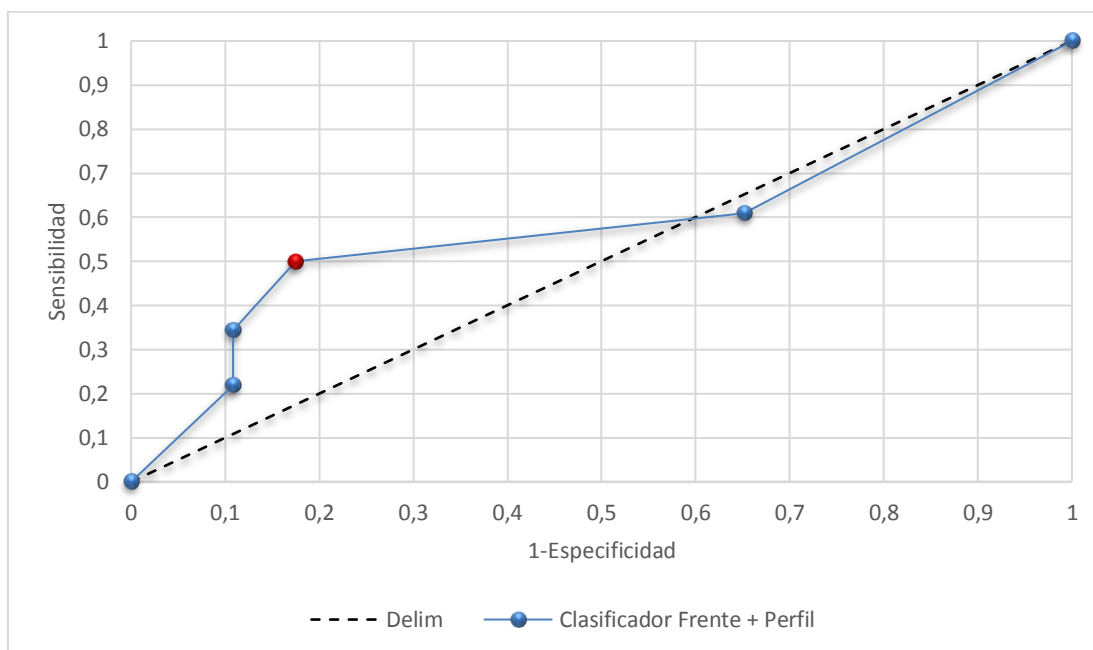


Ilustración 5-17 Banco de Pruebas: **Evaluación de los diferentes niveles de umbral para el clasificador de detección de ciclistas de perfil en función de la Sensibilidad y la Especificidad**. Para la evaluación de cada clasificador se empleó el banco de imágenes de prueba correspondiente y 4 valores de umbral distintos. La línea delimitadora diagonal representa el límite de aleatoriedad para la evaluación del clasificador. El punto rojo indica el mejor resultado obtenido.

La Ilustración 5-17 muestra valores muy parecidos de curva para los distintos valores de umbral. Siendo el mejor valor el de 0.75 (marcado en rojo). El análisis visual de la curva muestra resultados peores que para los anteriores clasificadores. Podemos concluir que este clasificador ha de ser mejorado para cumplir con la calidad de detección deseada para esta aplicación.

Por último, en la Tabla 5-8 se muestra un resumen de las áreas bajo la curva (AUC) para los distintos clasificadores.

	AUC
Clasificador Mixto	0,58
Clasificador Perfil	0,93
Clasificador Frente	0,81

Tabla 5-8 Valores AOC obtenidos.

De los resultados obtenidos se deduce que por un lado se ha de desechar la opción de utilizar el clasificador mixto ya que su valor de AUC es de 0.58 muy próximo a la diagonal de aleatoriedad (AUC=0.5).

Por otro lado, es obvio que se ha de utilizar el clasificador de perfil con un valor de umbral de 0.5 y el clasificador de frente con un valor de umbral de 0.42. Estos dos últimos clasificadores, por separado han tenido resultados óptimos para la detección de ciclistas.

#### 5.4 BANCO DE PRUEBAS PARA EL SEGUIMIENTO.

El banco de pruebas descrito en este apartado tiene dos grandes objetivos. El primero de ellos consiste en comprobar la robustez del algoritmo de seguimiento basado en el filtro de Kalman aplicado sobre el detector de ciclistas. Como segundo gran objetivo se encontraba comprobar la fiabilidad y precisión del sistema de estimación de distancias.

Para ello se diseñó un banco de pruebas consistente en la realización de una serie de trayectorias controladas en dos ejes (x e y). El dispositivo móvil se colocó en el punto (0,0) y la distribución de las trayectorias que se llevaron a cabo se muestra en la Ilustración 5-18

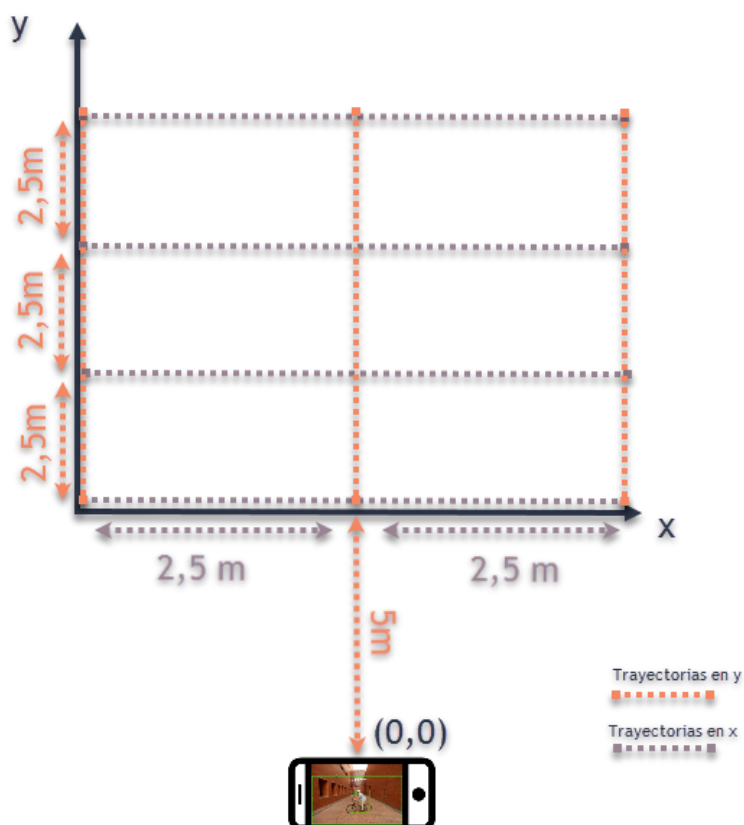


Ilustración 5-18 Definición de trayectorias en ambos ejes.

Para las pruebas se realizaron cuatro trayectorias a lo largo del eje x variando la distancia con respecto a el eje y en cada una de ellas, con el fin de evaluar el seguimiento a distintas

distancias y además se evaluó el sistema de estimación de distancias en el eje y, ya que en la ejecución de dichas trayectorias este valor debe permanecer prácticamente invariable.

Por otro lado, se realizaron siguiendo el mismo procedimiento, tres trayectorias a lo largo del eje y, variando únicamente la distancia con respecto al eje x. Con dicha prueba conseguimos evaluar tanto el seguimiento como el sistema de estimación de distancias a lo largo del eje x que en esta ocasión permanece invariable en toda la trayectoria.

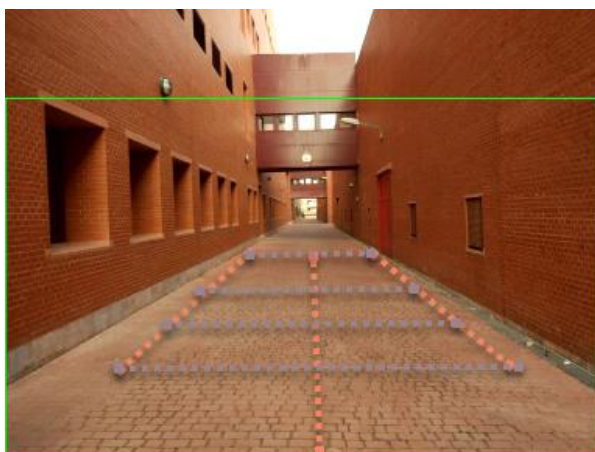


Ilustración 5-19 Definición real de trayectorias.

Cabe destacar que para la ejecución de dichas pruebas se realizaron 2 pasadas en ambos sentidos con dos ciclistas distintos para cada una de las trayectorias anteriormente citadas. Para el registro de datos se habilitó en la aplicación la opción “Capturar Frames” y la opción “Registrar datos”. Todos los frames capturados se les nombró con una marca horaria con la finalidad de poder tener una trazabilidad de los objetos detectados en el archivo de registro de datos. A continuación, se muestra a modo de ejemplo un extracto de las imágenes capturadas junto con su registro de datos.



Marca Horaria	Id Tracking	U	V	X	Y
18_7_5	ID:600	272	146	2,245484591	6,837219715
18_7_5	ID:600	262	148	2,078330755	7,024365902
18_7_5	ID:600	253	143	1,879384041	6,757407665
18_7_5	ID:600	246	141	1,78446877	6,932376385
18_7_5	ID:600	239	140	1,66301465	7,025844574

Ilustración 5-20 Banco de Pruebas: **Registro de Datos y Captura de Frames**. Para la evaluación del seguimiento y el sistema de estimación de distancias se utilizaron los frames capturados durante las pruebas y un registro de datos. El registro de datos contiene los siguientes campos:

- 1) Marca horaria con la finalidad de trazar sobre que objetos se realiza el seguimiento.
- 2) Id Tracking. Si el id tracking variara durante el periodo de seguimiento significaría que el filtro ha perdido el objeto y lo ha tratado como uno nuevo. En las capturas se apreciaría un cambio de color en la caja.
- 3) Valores U y V, informan del desplazamiento en cada eje en pixels.
- 4) Valores X e Y, informan del desplazamiento en cada eje en metros

#### 5.4.1 Resultados para el Seguimiento.

La Tabla 5-9 muestra los resultados de la distancia media detectada, la desviación típica y la varianza para cada una de las trayectorias bajo prueba. Hay que puntualizar que para los resultados de la estimación del desplazamiento a lo largo del eje y (bloqueo de x) se ha detectado un desplazamiento a la derecha en los resultados de 0,5 m. Esto se debe a un error de posicionamiento del sensor de la cámara que se desplazó levemente hacia la derecha. Los resultados de estimación de distancia en x han sido muy satisfactorios tanto para el cálculo de la posición media, como en la repetitividad de los mismos, obteniendo unos valores de desviación típica y varianza realmente bajos.

En cuanto a los desplazamientos a lo largo del eje horizontal (bloqueo del eje vertical) cabe destacar que la estimación de distancia es precisa a distancias cortas, pero va acumulando el error a medida que aumenta la distancia. La principal causa del error de estimación de distancias a lo largo del eje vertical se debe a la variación del recuadro que envuelve al objeto. Al aumentar la distancia en el eje, el objeto a detectar es más pequeño y la resolución a la que trabaja el sistema provoca que el desplazamiento del recuadro de los píxeles suponga un error mayor. Como es evidente este tipo de error disminuye a medida que el objeto se acerca al sensor del dispositivo hasta llegar a ser prácticamente nulo cuando la distancia se aproxima a los 4,5 metros.

Cabe destacar nuevamente la repetitividad del sistema a la hora de estimar distancias es muy fiable gracias a los datos obtenidos de desviación típica y varianza al ser ambos muy bajos.

Distancia(m)	$\bar{x} \text{ (m)}$	$\sigma(x) \text{ (m)}$	$\sigma_x^2 \text{ (m)}$
2,5	2,0917503	0,14316505	0,020496231
0	0,48739413	0,2103851	0,044261889
2,5	3,08173875	0,5940303	0,352871997
Distancia(m)	$\bar{y} \text{ (m)}$	$\sigma(y) \text{ (m)}$	$\sigma_y^2 \text{ (m)}$
5	6,758654253	0,314502108	0,098911576
7,5	11,03409286	0,585915214	0,343296638
10	15,73429503	0,621904535	0,386765251
12,5	22,15040487	0,660522273	0,436289673

Tabla 5-9 Distancia media. Desviación típica y varianza.

A continuación, se muestran las gráficas para distintas trayectorias en la que se indica la comparación entre la trayectoria real y la estimada por la aplicación. Adicionalmente, hemos representado las gráficas de error para cada de las comparaciones de trayectorias reales vs estimadas.

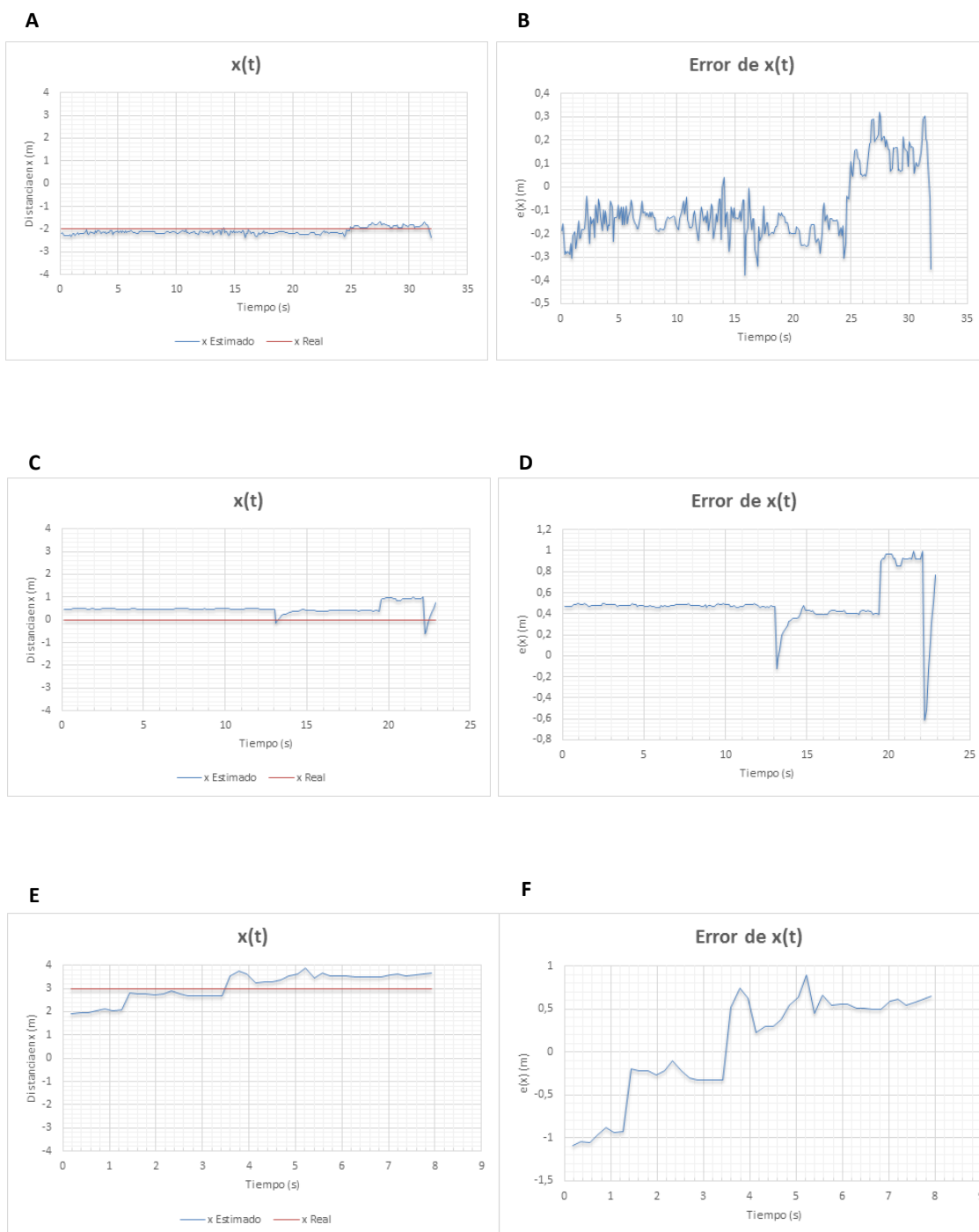
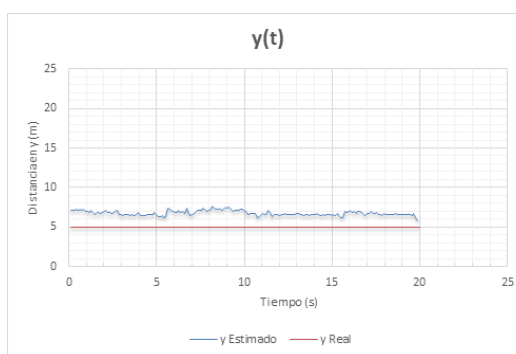


Ilustración 5-21 Estimación de distancias a lo largo del eje  $x$  y sus respectivos errores

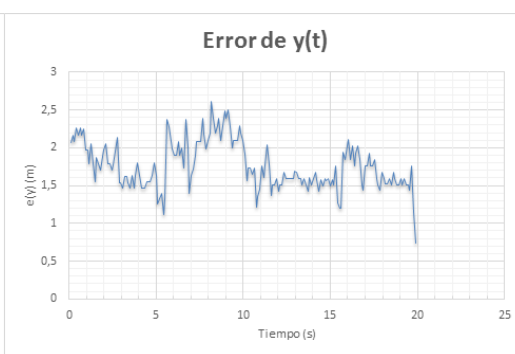
- A) Trayectoria  $x = -2$
- B) Error para Trayectoria  $x = -2$
- C) Trayectoria  $x = 0.5$
- D) Error para Trayectoria  $x = 0.5$
- E) Trayectoria  $x = 3$
- F) Error para Trayectoria  $x = 3$



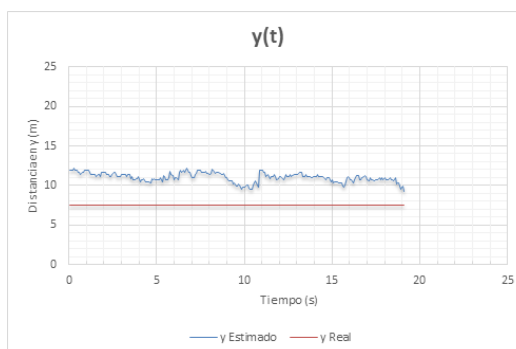
**A**



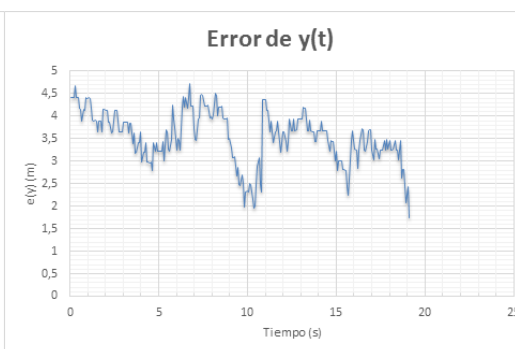
**B**



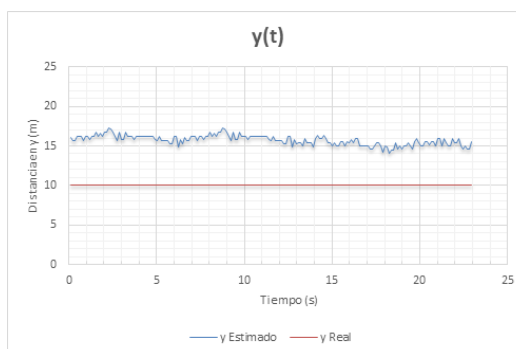
**C**



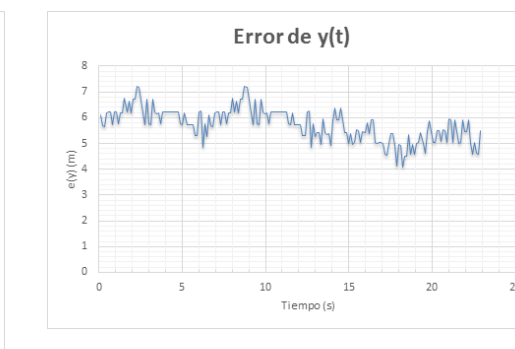
**D**



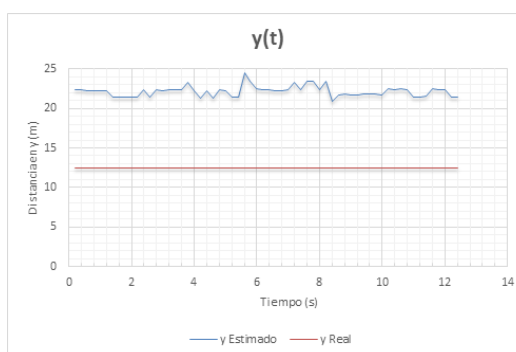
**E**



**F**



**G**



**H**

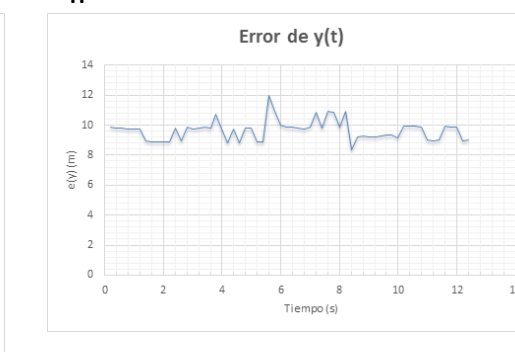


Ilustración 5-22 Estimación de distancias a lo largo del eje y y sus respectivos errores

- A) Trayectoria  $y=5$
- B) Error para Trayectoria  $y=5$
- C) Trayectoria  $y=7.5$
- D) Error para Trayectoria  $y=7.5$
- E) Trayectoria  $y=10$
- F) Error para Trayectoria  $y=10$
- G) Trayectoria  $y=12.5$
- H) Error para Trayectoria  $y=12.5$

En todas las gráficas se observa una desviación entre la trayectoria real y la trayectoria estimada. Dicha desviación viene ocasionada por varios motivos. El motivo principal es el caso anteriormente citado de variación del rectángulo envolvente al ciclista a lo largo de varios frames consecutivos. Otro claro caso de desviación viene provocado por el error humano cometido por el sujeto de pruebas ya que éste debería mantener una trayectoria ideal lo cual es prácticamente imposible debido a la escasa velocidad que requería la ejecución de la prueba. Para eliminar este error sería necesario que, durante la ejecución de la prueba, el sujeto se desplace a velocidad constante a lo largo de la trayectoria y a su vez modelar correctamente la matriz de covarianzas del error que utiliza dicho filtro para minimizar este tipo de error. A estos errores habría que sumar las imperfecciones del terreno o los errores derivados de la medida extraída del acelerómetro del dispositivo.

Por último, en la Tabla 5-10 se muestra la información ampliada en cuanto a errores medios por cada una de las trayectorias y las tasas de detección en función de los frames obtenidos y el número de falsos negativos obtenidos durante el ensayo.

TRAYECTORIA	FRAMES	FN	RATIO	FP	$\bar{e}$	$\overline{ABS}(e)$	$e_{rms}$
$x=-2,5$	47	1	97,87	0	0,082	0,522	0,593
$x=0$	37	2	94,59	0	0,487	0,108	0,531
$x=2,5$	27	2	92,59	0	0,092	0,111	0,170
$y=5$	39	0	100	0	1,723	0,288	1,790
$y=7,5$	38	1	97,37	0	3,535	0,466	3,582
$y=10$	32	0	100	0	11,52	5,73	13,285
$y=12,5$	21	6	71,42	0	9,650	4,005	9,673

Tabla 5-10 Errores y Ratios de Detección.

#### **5.4.2 Conclusiones para el seguimiento**

A la vista de los resultados del apartado anterior, se puede concluir que los resultados de las pruebas han sido satisfactorios por varios motivos.

- 1) La tasa de detección ha sido muy elevada para cualquiera de las trayectorias descritas dando una fiabilidad muy alta al sistema al reducir a 0 el número de falsos positivos obtenidos y teniendo unos valores prácticamente nulos de falsos negativos.
- 2) Las trayectorias descritas a lo largo del eje vertical han dado unos resultados de estimación de distancia a lo largo del eje horizontal muy precisos y con una repetitividad muy alta la cual otorga robustez al sistema para este tipo de estimación.
- 3) La estimación de distancias a lo largo del eje vertical es más exacta a distancias cortas, disminuyendo dicha exactitud a medida que nos alejamos del dispositivo. Aunque en este aspecto la aplicación es mejorable, para los casos de uso de nuestro sistema es más que suficiente el resultado obtenido ya que este error es subsanable implementando en el sistema un margen de seguridad más amplio para las distancias más alejadas. En futuras versiones se buscará acotar mejor la caja que envuelve al ciclista, para permitir su óptima identificación evitando errores de detección.

## 6. PRESUPUESTO

### 6.1 MANO DE OBRA

La mano de obra se divide en los costes por mano de obra de Ingeniería y los costes por mano de obra en Management o de Supervisión. Las horas de ingeniería atendiendo estas a tareas de desarrollo y documentales y las horas de Management a las propias de la naturaleza de gestión y dirección del desarrollo. A continuación, se muestra un desglose de dichos costes.

Tipo de Mano de Obra	Cantidad [Horas]	Coste Unitario [€/h]	Coste Total [€]
Ingeniería	360	15	5400
Management	20	30	600
Total			6000

Tabla 6-1 Costes por Mano de Obra.

### 6.2 MATERIAL

En las siguientes líneas se especifican los costes asociados al material requerido para la ejecución del desarrollo y pruebas de proyecto.

Material	Concepto	Cantidad	Coste Total [€]
Portátil LG	Ordenador Portatil	1	320
Xiaomi MI 5	Smartphone Pruebas	1	150
Xiaomi Redmi 3	Smartphone Pruebas	1	100
Total			570

Tabla 6-2 Costes por Materiales.

### 6.3 LICENCIAS DE SOFTWARE

A continuación, se detallan los costes derivados de todo el software licenciado necesario para la ejecución del proyecto.

Licencia de Software	Tipo	Cantidad	Coste Unitario [€]	Coste Total [€]
SvmLight	Gratuita	1	0	0
Android Studio	Gratuita	1	0	0
Microsoft Windows 7	Licencia de Estudiante	1	140	140
Microsoft Office 365	Licencia de Estudiante	1	140	140
Total			280	280

Tabla 6-3 Costes por Licencias Software.

### 6.4 PRESUPUESTO FINAL

Por último, se detalla un resumen de los costes totales en los que ha incurrido el proyecto.

Tipo de Coste	Coste Total [€]
Mano de Obra	6000
Material	570
Licencias de Software	280
Total	6850

Tabla 6-4 Costes totales.

## **7. CONCLUSIONES**

Tras haber alcanzado el objetivo principal del proyecto consistente en la carga de varios clasificadores SVM con distintas características y haber realizado la evaluación de dichos clasificadores tanto para la detección y el seguimiento, podemos concluir que el sistema es capaz de implementar dichos clasificadores de manera fiable en el sistema.

Concretando, de la evaluación de los clasificadores, podemos concluir que el clasificador mixto no es adecuado para la detección de ciclistas puesto que los valores obtenidos de las pruebas concluyen que la respuesta del mismo es totalmente aleatoria. Este resultado, aunque no era el esperado, es congruente con los fundamentos del detector utilizado ya que hay una gran diferencia en cuanto a las características obtenidas de una imagen de un ciclista de frente y otro de perfil. Dado que para los ciclistas de perfil la característica principal en cuanto al contorno se obtendrá de la forma de las ruedas y para los ciclistas de frente se obtendrá a partir del bloque bicicleta-persona, es posible que tal disparidad afecte a la hora de construir un detector válido.

Por el contrario, a la vista de los resultados en las pruebas para los detectores de frente y de perfil, se deduce que dichos clasificadores son aptos para la detección obteniendo mejores resultados la detección de perfil con respecto a la detección de ciclistas de frente.

En cuanto a las pruebas de seguimiento efectuadas, se concluye en base a los resultados obtenidos que éste se realiza de manera satisfactoria. Dicha conclusión se sostiene en base a la elevada tasa de detección y la alta repetitividad en cuanto a la estimación de distancias.

## **8. TRABAJOS FUTUROS**

A tenor de los resultados obtenidos con el detector mixto, para una futura versión de la aplicación se utilizará el clasificador para ciclistas de frente y un clasificador para peatones de manera conjunta. El contorno de un ciclista de frente y de un peatón se aproxima bastante, si se aúnan dichos clasificadores se obtendría en una sola aplicación tanto la detección de peatones como la detección de ciclistas en cualquier perspectiva, siendo dicha funcionalidad muy interesante de aplicar en el contexto de ADAS y Seguridad Vial.

Otro punto a mejorar en la versión futura de la aplicación sería la estimación de distancias a lo largo del eje y. Dicha mejora consistiría en la implementación de un margen de seguridad más amplio para las distancias más alejadas y reducir el error de la misma acotando mejor la caja que envuelve a la entidad detectada.

## 9. REFERENCIAS

- [1] J. Greenough, «<http://www.businessinsider.com/>,» Business Insider, 25 04 2016. [En línea]. Available: <http://www.businessinsider.com/report-10-million-self-driving-cars-will-be-on-the-road-by-2020-2016-4-25>. [Último acceso: 21 08 2017].
- [2] Organización Mundial de la Salud, «Informe mundial sobre prevención de los traumatismos causados por el tránsito,» Margaret Peden ... [y otros], Ginebra, 2004.
- [3] Dirección General de Trafico, «Las principales cifras de la siniestralidad Vial España 2015,» DGT, 2015.
- [4] Dirección General de Tráfico, «<http://www.lamoncloa.gob.es/>,» 2011. [En línea]. Available: [http://www.lamoncloa.gob.es/espana/eh15/seguridad/Documents/estrategico\\_2020\\_004.pdf](http://www.lamoncloa.gob.es/espana/eh15/seguridad/Documents/estrategico_2020_004.pdf). [Último acceso: 21 08 2017].
- [5] Fundación Instituto Tecnológico para la Seguridad del Automóvil, «<https://stopaccidentes.org/>,» 2008. [En línea]. Available: [https://stopaccidentes.org/uploads/file/Costes\\_accidentes.pdf](https://stopaccidentes.org/uploads/file/Costes_accidentes.pdf). [Último acceso: 21 08 2017].
- [6] A. Kesting, M. Treiber, M. Schönhof y D. Helbing, «Adaptive cruise control design for active congestion avoidance,» *Transportation Research Part C: Emerging Technologies*, vol. 16, nº 6, pp. 668-683, 2008.
- [7] X. Li, J. Cui, S. An y M. Parsafard, «Stop-and-go traffic analysis: Theoretical properties, environmental impacts and oscillation mitigation,» *Transportation Research Part B: Methodological*, vol. 70, pp. 319-339, 2014.
- [8] CARICOS, «<http://www.caricos.com/>,» 2015. [En línea]. Available: [http://www.caricos.com/cars/v/vw/2015\\_volkswagen\\_touareg/images/43.html](http://www.caricos.com/cars/v/vw/2015_volkswagen_touareg/images/43.html). [Último acceso: 12 09 2017].
- [9] E. Cano, «Asistente predictivo, para ahorrar aún más,» *ABC*, 31 08 2015.
- [10] Ó. Cisneros, «<http://www.centro-zaragoza.com/>,» 09 2009. [En línea]. Available: [http://www.centro-zaragoza.com:8080/web/sala\\_prensa/revista\\_tecnica/hemeroteca/articulos/R41\\_A7.pdf](http://www.centro-zaragoza.com:8080/web/sala_prensa/revista_tecnica/hemeroteca/articulos/R41_A7.pdf). [Último acceso: 2017 08 21].
- [11] A. Mukhtar, L. X. a. T y B. Tang, «Vehicle Detection Techniques for Collision Avoidance Systems: A Review,» *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, nº 5, pp. 2318-2338, 2015.
- [12] BMWGroup, «<https://www.press.bmwgroup.com/>,» BMWGroup, 16 05 2011. [En línea]. Available: <https://www.press.bmwgroup.com/global/article/detail/T0108975EN/the-left-turn-assistant-%E2%80%93turning-left-safely-by-looking-out-for-oncoming-traffic?language=en>. [Último acceso: 2017 08 21].
- [13] <http://www.revistacesvimap.com>, «<http://www.revistacesvimap.com>,» 29 10 2015. [En línea].



Available: <http://www.revistacesvimap.com/audi-q7-quattro/>. [Último acceso: 12 09 2017].

- [14] National Highway Traffic Safety Administration, «<https://www.safercar.gov/>,» National Highway Traffic Safety Administration, [En línea]. Available: <https://www.safercar.gov/Vehicle-Shoppers/Safety-Technology/AEB/aeb>. [Último acceso: 2017 08 21].
- [15] DMV, «[www.dmv.com](http://www.dmv.com/),» 02 02 2015. [En línea]. Available: <https://www.dmv.com/blog/automatic-braking-technology-to-be-added-on-NHTSAs-recommended-safety-tech-list-521407>. [Último acceso: 12 09 2017].
- [16] Movon Corporation, «[http://www.mdas.co.kr](http://www.mdas.co.kr/),» Movon Corporation, [En línea]. Available: [http://www.mdas.co.kr/eng/technology/technology1\\_2.php#lside](http://www.mdas.co.kr/eng/technology/technology1_2.php#lside). [Último acceso: 21 08 2017].
- [17] MOVON, «<http://mdas.co.kr/>,» [En línea]. Available: <http://mdas.co.kr/eng/product/automobile-safety-devices.php>. [Último acceso: 12 09 2017].
- [18] Apkpure, «[https://apkpure.com](https://apkpure.com/),» [En línea]. Available: <https://apkpure.com/acodriver-5/com.evotegra.aCoDriver>. [Último acceso: 12 09 2017].
- [19] Google, «Google Play,» iOnRoad, [En línea]. Available: <https://play.google.com/store/apps/details?id=com.picitup.iOnRoad&hl=es>. [Último acceso: 09 10 2017].
- [20] P. Viola y M. J. Jones, «Robust Real-Time Face Detection,» *International Journal of Computer Vision*, vol. 57, nº 2, p. 137–154, 2004.
- [21] D. G. Lowe, «Object recognition from local scale-invariant features,» *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150-1157, 1999.
- [22] H. Bay, T. Tuytelaars y L. V. Gool, «SURF: Speeded Up Robust Features,» *Computer Vision -- ECCV 2006: 9th European Conference on Computer Vision*, pp. 404-417, 2006.
- [23] N. Dalal y B. Triggs, «Histograms of oriented gradients for human detection,» *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886-893, 2005.
- [24] B. W. Silverman y M. C. Jones, «An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges,» *International Statistical Review / Revue Internationale De Statistique*, vol. 57, nº 3, p. 233–238, 1951.
- [25] Zhang, S. a. Li, X. a. Zong, M. a. Zhu, X. a. Cheng y Debo, «Learning K for kNN Classification,» *ACM Trans. Intell. Syst. Technol*, vol. 8, nº 3, pp. 43:1--43:19, 2017.
- [26] O. Chapelle, P. Haffner y V. N. Vapnik, «Support vector machines for histogram-based image classification,» *IEEE Transactions on Neural Networks*, vol. 10, nº 5, pp. 1055-1064, 2010.
- [27] A. R. López, *Implementación de Algoritmos de Visión por Computador en Plataforma Android*, uc3m Léguas, 2013.

- [28] D. Martín, F. García, B. Musleh, D. Olmeda, G. Peláez, P. Marín, A. Ponz, C. Rodríguez, A. Al-Kaff, A. d. I. Escalera y J. Armingol, «IVVI 2.0: An intelligent vehicle based on computational perception,» *Expert Systems with Applications*, vol. 41, nº 17, pp. 7927-7944, 2014.
- [29] OpenCV team, «<http://opencv.org>,» OpenCV team, [En línea]. Available: <http://opencv.org/about.html>. [Último acceso: 22 08 2017].
- [30] Google INC, «<https://developer.android.com>,» Google INC, [En línea]. Available: <https://developer.android.com/studio/index.html>. [Último acceso: 2017 08 22].
- [31] T. Joachims, *Learning to classify text using support vector machines*, Boston: Kluwer Academic, 2002.
- [32] C. Cortes y V. Vapnik, «Support-vector networks,» *Machine Learning*, vol. 20, nº 3, p. 273–297, 1995.
- [33] R. Kalman, «A New Approach to Linear Filtering and Prediction Problems,» *Elgar Reference Collection. International Library of Critical Writings in Econometrics*, vol. 5, pp. 133-43, 1994.
- [34] C. Huang, B. Wu y R. Nevatia, «Robust Object Tracking by Hierarchical Association of Detection Responses,» *European Conference on Computer Vision*, pp. 788-801, 2008.
- [35] V. Cancho Arnesto, «<https://e-archivo.uc3m.es/>,» 02 10 2014. [En línea]. Available: [https://e-archivo.uc3m.es/bitstream/handle/10016/24754/TFG\\_Victor\\_Cancho\\_Arnesto\\_2014.pdf?sequence=1&isAllowed=y](https://e-archivo.uc3m.es/bitstream/handle/10016/24754/TFG_Victor_Cancho_Arnesto_2014.pdf?sequence=1&isAllowed=y). [Último acceso: 13 09 2017].
- [36] J. Urdiales de la Parra, «[e-archivo.uc3m.es](https://e-archivo.uc3m.es/),» 07 03 2016. [En línea]. Available: [https://e-archivo.uc3m.es/bitstream/handle/10016/24111/TFG\\_Jesus\\_Urdiales\\_de-la-Parra\\_2016.pdf](https://e-archivo.uc3m.es/bitstream/handle/10016/24111/TFG_Jesus_Urdiales_de-la-Parra_2016.pdf). [Último acceso: 13 09 2017].
- [37] A. d. I. Escalera Hueso, *Visión por computador : fundamentos y métodos*, Madrid: Prentice Hall, 2001.
- [38] JKalman, «[sourceforge.net](https://sourceforge.net),» 14 03 2014. [En línea]. Available: <https://sourceforge.net/projects/jkalman/>. [Último acceso: 13 09 2017].
- [39] A. P. Bradley, «The use of the area under the ROC curve in the evaluation of machine learning algorithms,» *Pattern recognition*, vol. 30, nº 7, pp. 1145-1159, 1997.
- [40] J. . Pérez, V. . Milanés, J. . Alonso, E. . Onieva y T. d. Pedro, «Adelantamiento con vehículos autónomos en carreteras de doble sentido,» *Revista Iberoamericana De Automatica E Informatica Industrial*, vol. 7, nº 3, pp. 25-33, Revista Iberoamericana De Automatica E Informatica Industrial 2010.